

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 1

ЗАТВЕРДЖЕНО

Науково-методичною радою
Державного університету
«Житомирська політехніка»

протокол № від _____ р.

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ для написання лабораторних робіт з навчальної дисципліни

«Розробка, безпека та тестування мобільних додатків»
для здобувачів вищої освіти освітнього ступеня «БАКАЛАВР»
спеціальності 122 «Комп'ютерні науки»
освітньо-професійна програма «Комп'ютерна графіка та розробка ігор»
факультет інформаційно-комп'ютерних технологій
кафедра комп'ютерних наук

Рекомендовано на засіданні
кафедри комп'ютерних наук
30 червня 2025 р.,
протокол № 7

Розробники: асистент кафедри комп'ютерних наук Денис ЛЮБЧЕНКО
асистент кафедри комп'ютерних наук Віктор ЖЕЛІЗКО
асистент кафедри комп'ютерних наук Андрій ОРИНЧАК

Житомир
2025

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 2

ЗМІСТ

ВСТУП	4
ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	6
Тема 1. Вступ до гібридної розробки	8
Тема 2. Архітектура мобільних додатків	12
ЛАБОРАТОРНА РОБОТА №1- Встановлення середовища розробки. Створення гібридного мобільного додатку (візитівки студента) на Ionic Framework	17
Тема 3. Створення гібридних додатків	20
Тема 4. Інтеграція нативних функцій у гібридних додатках	25
ЛАБОРАТОРНА РОБОТА №2 - Створення ToDo-застосунку. Локальне збереження даних із використанням Ionic Storage	31
Тема 5. Інтеграція з Firebase та робота з даними в гібридних додатках	34
Тема 6. Використання офлайн баз даних у мобільних додатках	41
ЛАБОРАТОРНА РОБОТА №3 - - Firebase у мобільних застосунках. Реалізація автентифікації та збереження даних користувача	47
Тема 7. Розробка користувацького інтерфейсу (UI/UX) у мобільних додатках	54
Тема 8. Використання Firebase для автентифікації та моніторингу у мобільних додатках	58
ЛАБОРАТОРНА РОБОТА №4- Використання нативних функцій у мобільних застосунках. Збереження файлів у хмарі	64
Тема 9. Розробка користувацького інтерфейсу у гібридних мобільних додатках	69
Тема 10. Просунутий мобільний UI/UX-дизайн	75
Тема 11. Оптимізація продуктивності гібридних мобільних додатків	80

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 3

Тема 12. Адаптація мобільного інтерфейсу під різні платформи та типи екранів	85
Тема 13. Тестування гібридних мобільних додатків	94
Тема 14. Безпека гібридних мобільних додатків	100
Тема 15. Розгортання гібридних мобільних додатків. CI/CD	105
ЛАБОРАТОРНА РОБОТА №5- Локалізація, геолокація, мапа та push-нотифікації в гібридних мобільних додатках	90
ЛАБОРАТОРНА РОБОТА №6 -Тестування, безпека та розгортання гібридного мобільного додатку	110
СПИСОК ЛІТЕРАТУРИ	115
ДОДАТКИ	116

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 4

ВСТУП

Мобільна розробка є однією з найдинамічніших галузей у сфері інформаційних технологій. В умовах зростання попиту на мобільні застосунки з боку бізнесу, держави та кінцевих користувачів, важливим є формування у здобувачів вищої освіти практичних навичок створення, тестування та забезпечення безпеки мобільних рішень.

Навчальна дисципліна «Розробка, безпека та тестування мобільних додатків» спрямована на формування у студентів навичок проєктування, розробки, забезпечення безпеки, тестування та презентації мобільних застосунків із використанням гібридних і нативних підходів. Особлива увага приділяється застосуванню сучасних фреймворків (Ionic Framework, Angular, Capacitor), хмарних сервісів (Firebase) і нативних можливостей мобільних платформ (Android/iOS).

Мета дисципліни

- Формування у студентів практичних навичок розробки мобільних додатків із використанням сучасних інструментів і технологій гібридної та нативної розробки, забезпечення їх безпеки та якісного тестування.
- Завдання дисципліни:
- Ознайомлення з мобільними платформами (iOS, Android) і різними підходами до розробки (гібридна/нативна).
- Опрацювання фреймворку Ionic для створення гібридних додатків.
- Розробка UI/UX-дизайну мобільного інтерфейсу.
- Вивчення основ нативної розробки (Android Studio, Xcode).
- Засвоєння принципів безпеки мобільних застосунків.
- Вивчення методів тестування: юніт-, інтеграційне, UI-тестування, тестування продуктивності.
- Реалізація та презентація власного мобільного додатку як завершального етапу.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 5

Очікувані результати навчання:

- Після завершення курсу здобувач зможе:
- Розробляти мобільні додатки з використанням Ionic, Capacitor, Firebase.
- Інтегрувати нативні функції (камера, геолокація, біометрія тощо).
- Реалізовувати багатомовність та адаптивний дизайн.
- Проводити тестування та забезпечувати захист даних.
- Публікувати застосунки, використовувати CI/CD.
- Працювати з API сторонніх сервісів.

Розвиток soft skills:

- Критичне мислення (вибір технологій, аналіз ризиків),
- Командна робота (групові проєкти),
- Креативність (UI/UX дизайн, інновації),
- Тайм-менеджмент (дедлайни, пріоритети),
- Комунікація (захист проєкту, командні обговорення),
- Гнучкість та здатність до навчання (нові API, інструменти).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 6

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. Основи гібридної мобільної розробки

Тема 1. Вступ до гібридної розробки. Розглядаються основні підходи до створення мобільних додатків: нативні, гібридні, веб-додатки. Аналізуються їх переваги та недоліки. Вивчаються приклади популярних застосунків, створених за допомогою гібридних фреймворків. Окреслюється екосистема мобільної розробки (iOS, Android, гібридна платформа).

Тема 2. Архітектура мобільних додатків. Висвітлюються основні компоненти сучасного мобільного застосунку. Розглядаються модульність, компонентний підхід та архітектурні рішення на прикладі фреймворків Ionic, React Native та Flutter.

Тема 3. Створення гібридних додатків. Покроково розглядається встановлення середовища розробки: Node.js, npm, Android Studio, Ionic CLI. Практичне завдання — створення першого мобільного застосунку в середовищі Ionic.

Тема 4. Інтеграція нативних функцій у гібридних додатках. Вивчається взаємодія з нативними API за допомогою Cordova та Capacitor. Практичне завдання — інтеграція камери, геолокації, акселерометра.

Тема 5. Інтеграція API та робота з даними. Висвітлюються принципи REST, робота з HTTP-запитами, форматами JSON і XML. Реалізується приклад отримання даних з відкритого API.

Тема 6. Використання офлайн баз даних у мобільних додатках. Порівнюються Local Storage, IndexedDB та SQLite. Окремо розглядається використання плагіну SQLite у проєктах Ionic.

Тема 7. Firebase як BaaS. Ознайомлення з Firebase як backend-сервісом. Вивчаються можливості Firestore, Realtime Database, Cloud Functions та Firebase Storage.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 7

Тема 8. Аутентифікація та моніторинг через Firebase. Практичне налаштування Firebase Authentication. Додатково — інтеграція Firebase Analytics та моніторинг активності користувачів.

Змістовий модуль 2. UI/UX і продуктивність гібридних додатків

Тема 9. Розробка користувацького інтерфейсу. Розглядаються принципи адаптивності, доступності та зручності користувацького інтерфейсу. Практична робота з компонентами Ionic.

Тема 10. Просунутий мобільний UI/UX-дизайн. Аналізуються сучасні тренди: матеріальний дизайн, інтерактивні елементи, кольори, типографіка. Вивчаються принципи створення інтуїтивно зрозумілого інтерфейсу.

Тема 11. Оптимізація продуктивності. Висвітлюються методи підвищення швидкодії: кешування, lazy loading, мінімізація запитів. Практичне використання Lighthouse, Firebase Performance Monitoring.

Тема 12. Адаптація під екрани та платформи. Проводиться оптимізація для різних типів пристроїв: телефони, планшети, Android, iOS. Тестування UI на різних екранах.

Змістовий модуль 3. Тестування, безпека та розгортання

Тема 13. Тестування гібридних додатків. Огляд інструментів автоматизованого тестування: Appium, Selenium. Ручне тестування через Chrome DevTools. Аналіз типових вразливостей.

Тема 14. Безпека мобільних додатків. Вивчаються основи захисту даних користувача, авторизації, інтеграції OAuth 2.0 та JWT. Практична реалізація засобів захисту на базі Firebase.

Тема 15. Розгортання мобільних додатків. Підготовка до публікації в Google Play та App Store. CI/CD через GitHub Actions або CircleCI. Налаштування Firebase Analytics та Crashlytics для відстеження роботи застосунку.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 8

Тема 16. Монетизація. Огляд моделей монетизації (реклама, підписка, in-app purchases). Практика інтеграції AdMob або банерної реклами в додаток.

Тема 1. Вступ до гібридної розробки

1.1. Сучасний контекст мобільної розробки

Упродовж останніх двох десятиліть мобільні застосунки стали невід’ємною частиною нашого повсякденного життя. Вони використовуються у всіх сферах: від комунікації до фінансів, від освіти до медицини. Через стрімке зростання попиту на мобільні сервіси виникає потреба у швидкій, масштабованій і зручній розробці застосунків, що працюють на різних платформах — Android, iOS, іноді ще й у вебі.

Розробка таких застосунків потребує вибору ефективної архітектури. Найпоширенішими підходами сьогодні є:

- **Нативна розробка** — створення окремих додатків для кожної платформи з використанням мов програмування та SDK, характерних для цієї ОС (Java/Kotlin для Android, Swift/Objective-C для iOS).
- **Гібридна розробка** — створення одного додатку з використанням веб-технологій (HTML, CSS, JS), який запускається у контейнері, що надає доступ до нативного функціоналу.
- **Прогресивні веб-додатки (PWA)** — вебсайти з можливістю офлайн-доступу, встановлення на головний екран і частковим доступом до функцій пристрою.

Кожен підхід має свої переваги та недоліки. Нативні додатки демонструють високу продуктивність і глибоку інтеграцію з пристроєм, проте їх розробка є дорогою, оскільки потребує двох окремих команд. PWA — швидкий у розробці

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 9

і гнучкий, але має обмежений доступ до нативного API. Гібридна розробка поєднує переваги обох підходів — вона дозволяє писати єдиний код для кількох платформ, забезпечує достатній рівень інтерактивності і швидке оновлення функціоналу.

1.2. Що таке гібридний мобільний додаток

Гібридні додатки — це мобільні застосунки, що створюються з використанням веб-технологій (HTML5, CSS3, JavaScript) і працюють у вбудованому WebView-контейнері. Завдяки цьому вони виглядають і функціонують як нативні додатки, але мають єдину кодову базу, яку можна запускати на різних платформах.

Такі додатки реалізуються за допомогою спеціальних фреймворків і платформ, які забезпечують:

- запуск веб-коду у контейнері (як правило, WebView),
- доступ до нативних можливостей пристрою (камера, Bluetooth, GPS тощо),
- створення зручного та адаптивного інтерфейсу.

Найпопулярніші гібридні фреймворки:

- **Ionic Framework** — використовує Angular, React або Vue як основу, надає набір компонентів UI, має CLI, і працює у зв'язці з Capacitor.
- **Apache Cordova** — старіший інструмент, який забезпечує доступ до нативних API, але менш активно підтримується.
- **React Native** (частково гібридний) — дозволяє писати логіку на JavaScript, але створює справжні нативні UI-компоненти.

1.3. Переваги та обмеження гібридної розробки

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 10

Переваги:

- **Швидка розробка:** одна кодова база — менше часу на підтримку і реалізацію функцій.
- **Кросплатформеність:** підтримка одразу кількох ОС.
- **Спрощений CI/CD:** єдине середовище для інтеграції, тестування і деплою.
- **Менше залежності від спеціалістів по нативній розробці.**

Недоліки:

- **Продуктивність:** у складних анімаціях або тривимірній графіці можуть виникати затримки.
- **Обмежений доступ до деяких API:** деякі нативні можливості доступні тільки через кастомні плагіни.
- **Дизайн-парадигми:** інтерфейс може не повністю відповідати гайдлайнам Android або iOS.

Однак завдяки сучасним рішенням (Capacitor, Ionic UI Components, Angular Material) ці недоліки часто зводяться до мінімуму.

1.4. Поняття екосистеми гібридної мобільної розробки

У гібридній розробці важливо не лише розмістити код у WebView, а й організувати повноцінне середовище розробки. У цьому контексті **екосистема** включає:

- **Фреймворк для створення інтерфейсу** (наприклад, Angular або React);
- **Компонентну бібліотеку для UI** (Ionic Components);
- **Середовище виконання на пристрої** (Capacitor — міст між WebView і нативними API);

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 11

- **Інструменти для розгортання і тестування** (Android Studio, Xcode, Firebase, Appium);
- **Служби бекенду** — Firebase, AWS Amplify, Supabase, тощо;
- **Інструменти монетизації** — Google AdMob, in-app purchases.

Особливу роль відіграє **Ionic CLI** — інструмент, що дозволяє створювати нові проєкти, запускати їх у браузері, збирати білди для Android та iOS, додавати платформи та плагіни.

1.5. Приклади застосування гібридної розробки

Гібридна розробка активно використовується великими компаніями. Наприклад:

- **Sworakit** — популярний фітнес-додаток, побудований на Ionic Framework.
- **MarketWatch** — додаток новин і біржових оглядів, що працює на гібридній основі.
- **Diesel, Untappd, Pacifica** — приклади бізнесів, які скоротили витрати і час виходу на ринок, обравши гібридний підхід.

Це підтверджує життєздатність гібридної технології в комерційних та високонавантажених проєктах.

1.6. Ключові навички, що формуються

Після засвоєння даної теми студент має вміти:

- відрізняти гібридні, нативні та PWA-додатки;
- аргументовано обирати платформу для розробки;
- пояснювати переваги і обмеження гібридного підходу;
- знати основні інструменти, фреймворки та компоненти екосистеми;
- налаштовувати середовище розробки Ionic Framework;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 12

Тема 2. Архітектура мобільних додатків

2.1. Значення архітектури у розробці

Архітектура програмного забезпечення — це фундамент, на якому будується весь застосунок. У контексті мобільних додатків, вона визначає структуру коду, розподіл відповідальностей, логіку взаємодії компонентів, а також впливає на тестованість, розширюваність і продуктивність системи. Особливо важливою є архітектура у гібридних мобільних проєктах, де поєднуються особливості веб-програмування і доступ до нативних функцій пристрою.

У мобільних додатках архітектура повинна забезпечувати:

- ефективне розділення логіки, інтерфейсу та даних;
- можливість повторного використання компонентів;
- швидку адаптацію до змін;
- безпечну взаємодію з API;
- стійкість до помилок і масштабованість.

Найбільш поширеними архітектурними підходами є **MVC (Model-View-Controller)**, **MVVM (Model-View-ViewModel)** та **Clean Architecture**. У рамках Angular, на якому базується Ionic, реалізація наближена до компонентно-сервісної архітектури, що поєднує елементи MVVM і модульності.

2.2. Структура гібридного мобільного додатку на базі Ionic

Типовий проєкт Ionic має чітку розбивку на модулі, компоненти, сервіси та сторінки. Основні складові:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 13

1. **AppModule** — головний модуль Angular-застосунку. Містить інформацію про компоненти, імпортовані модулі, провайдери сервісів.
2. **AppComponent** — кореневий компонент, з якого запускається додаток. Тут ініціалізується навігація, зчитуються глобальні дані тощо.
3. **Pages (Сторінки)** — окремі розділи UI (наприклад, HomePage, LoginPage). Кожна сторінка складається з HTML-шаблону, CSS-стилів і TypeScript-класу.
4. **Services (Сервіси)** — класи, що реалізують бізнес-логіку, роботу з API, обробку даних (наприклад, TaskService, AuthService). Сервіси зазвичай є singleton і інжектуються через dependency injection.
5. **Models (Моделі)** — інтерфейси або класи, що описують структуру об'єктів (Task, User, Product). Вони дозволяють типізувати дані, уникати помилок та покращити читабельність коду.
6. **Routing (Маршрутизація)** — у файлі app-routing.module.ts налаштовується навігація між сторінками. Angular Router дозволяє використовувати параметри маршруту, lazy loading, guard-и для авторизації.
7. **Shared Module** — загальний модуль, де зберігаються повторно використовувані компоненти (наприклад, кнопки, хедери, модальні вікна).
8. **Environment** — змінні середовища, API-ключі та конфігурація Firebase або іншого бекенду.

2.3. Компоненти архітектури та їх призначення

View (Представлення)

HTML-шаблони разом зі стилями формують інтерфейс користувача. За допомогою Angular-розмітки (*ngIf, [(ngModel)], *ngFor) створюються

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 14

адаптивні, інтерактивні елементи. Компоненти Ionic значно спрощують реалізацію кнопок, списків, навігаційних панелей, форм.

Component/Controller (Компонент логіки)

TypeScript-клас сторінки виконує роль контролера — обробляє події, перевіряє дані, викликає сервіси, передає дані у View. Наприклад, метод addTask() у HomePage звертається до TaskService, передає нове завдання і перезавантажує список.

Service (Сервіс логіки)

Реалізує логіку збереження, обробки або передачі даних. Наприклад, AuthService виконує реєстрацію користувача через Firebase, TaskService опрацьовує список справ і працює з локальним або хмарним сховищем.

Model (Модель даних)

Забезпечує чітку структуру. Наприклад:

```
export interface Task {
  id: string;
  title: string;
  description?: string;
  dueDate?: string;
  isDone: boolean;
}
```

Guard-и та Pipes. Guard-и (наприклад, AuthGuard) захищають маршрути. Pipes (наприклад, DatePipe) дозволяють обробляти дані прямо у шаблоні.

2.4. Архітектурні патерни у гібридній розробці

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 15

У складних додатках, особливо коли йдеться про спільну роботу кількох розробників, важливо дотримуватись патернів. Найчастіше в гібридній розробці з Angular використовуються такі підходи:

- **MVVM (Model-View-ViewModel):**

Model — інтерфейси (Task), View — HTML-шаблон, ViewModel — TypeScript-компонент. Дозволяє відокремити логіку від відображення.

- **Reactive programming (RxJS):**

Сервіси повертають Observable, компоненти підписуються і реагують на зміни даних. Це важливо для взаємодії з Firebase, HTTP-запитів, реакції на події.

- **Lazy loading:**

Застосовується для оптимізації продуктивності. Завантаження модулів лише під час переходу на певну сторінку зменшує час старту додатку.

- **Dependency Injection (DI):**

Angular автоматично передає залежності (сервіси) в компоненти. Це дозволяє зручно тестувати й масштабувати застосунок.

2.5. Підключення нативних функцій: роль Capacitor

Гібридні додатки завдяки Capacitor мають змогу використовувати функції пристрою так, ніби це нативна розробка. Capacitor працює як міст між WebView (у якому виконується код Angular) та SDK пристрою (Java/Swift).

Capacitor дозволяє:

- підключати камеру, галерею, сканер;
- отримувати координати (геолокація);
- працювати з біометрією;
- показувати push-нотифікації;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 16

- керувати файлами, мережами, сховищем.

Підключення здійснюється через NPM і виклики в Angular-сервісах:

```
import { Camera } from '@capacitor/camera';

const image = await Camera.getPhoto({
  quality: 90,
  allowEditing: false,
  resultType: CameraResultType.Base64
});
```

Це робить архітектуру гібридного додатку більш універсальною: можна за потреби реалізовувати частину функцій у нативному коді (Java/Swift), не змінюючи основну логіку Angular.

2.6. Організація папок у проєкті

Для підтримки читабельності та масштабованості проєкту рекомендується така структура:

```
src/
├── app/
│   ├── pages/           ── сторінки (Home, Login)
│   ├── services/        ── сервіси
│   ├── models/          ── інтерфейси
│   ├── shared/           ── спільні компоненти
│   ├── guards/          ── захист маршрутів
│   └── app-routing.module.ts
├── assets/
└── environments/
```

Це дає змогу відокремити зони відповідальності, прискорює навігацію по коду і полегшує внесення змін.

2.7. Висновки

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 17

Розуміння архітектури мобільного застосунку є ключовим етапом у підготовці професійного розробника. Добре спроектована структура дозволяє:

- швидко масштабувати застосунок,
- мінімізувати кількість помилок,
- покращити продуктивність,
- ефективно управляти станом і даними.

Застосування Angular, Ionic і Capacitor у поєднанні з сучасними патернами дозволяє створювати стабільні, гнучкі та зручні у супроводі гібридні додатки, які відповідають очікуванням користувачів і вимогам бізнесу.

ЛАБОРАТОРНА РОБОТА №1

Тема: Встановлення середовища розробки. Створення гібридного мобільного додатку (візитівки студента) на Ionic Framework

Мета роботи: Набути практичних навичок встановлення та налаштування середовища гібридної мобільної розробки з використанням Ionic Framework, створення і запуску простого застосунку, запуску симулятора Android, а також ознайомлення зі структурою проекту та основними командами CLI.

Короткі теоретичні відомості:

Ionic Framework — популярний open-source фреймворк для розробки гібридних мобільних застосунків із використанням веб-технологій (HTML, CSS, JS/TS). У якості базового фронтенду може використовуватись Angular, React або Vue. У цій роботі використовується зв'язка Ionic + Angular.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 18

Capacitor — інструмент для доступу до нативних можливостей пристрою. Він дозволяє компілювати додаток для Android та iOS, підключати камеру, Bluetooth, геолокацію тощо.

Node.js та **npm** — необхідні для керування залежностями та встановлення CLI.

Android Studio — офіційне середовище розробки для Android, яке містить емулятор пристроїв (AVD), SDK, та інші необхідні інструменти для збірки проєктів.

Зміст роботи

Завдання 1. Перевірка наявності Node.js

У терміналі виконайте:

```
node -v
```

Якщо версія нижча за 21 — оновіть Node.js через [офіційний сайт](#).

Завдання 2. Встановлення Ionic CLI

У глобальному режимі:

```
npm install -g @ionic/cli
```

Увага: У Windows слід відкривати саме **Command Prompt (cmd)** або **термінал IDE**, а не PowerShell.

Завдання 3. Створення нового проєкту

У командному рядку:

```
ionic start
```

Під час запуску:

- Назва проєкту: student-card
- Framework: Angular
- Starter Template: blank

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 19

- Оберіть NgModules

Перейдіть у створену директорію:

```
cd student-card
```

Запустіть застосунок у браузері:

```
ionic serve
```

Завдання 4. Запуск на Android через емулятор

1. Встановіть Android Studio.
2. Відкрийте **Device Manager** та створіть AVD.
3. Увімкніть **апаратну віртуалізацію** у BIOS, якщо AVD не запускається.
4. Встановіть Android-платформу до проєкту:

```
npm install @capacitor/android
ionic cap add android
ionic capacitor sync
```

5. Відкрийте проєкт в Android Studio:

```
ionic cap open android
```

6. Запустіть проєкт на емуляторі або реальному пристрої.

Завдання 5. Налаштування Git-репозиторію

1. Створіть новий репозиторій на git.ztu.edu.ua
2. Додайте всі файли проєкту до Git:

```
git init
git add .
git commit -m "lab1"
```

3. Під'єднайте віддалений репозиторій та виконайте push.
4. Додайте викладача до репозиторію.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 20

Контрольні запитання:

1. Які основні відмінності між гібридним, нативним і веб-додатком?
2. Які можливості надає Capacitor у порівнянні з Cordova?
3. У чому перевага використання Ionic CLI для початку нового проєкту?
4. Як працює команда ionic capacitor sync і чим вона відрізняється від ionic serve?
5. Як запустити застосунок на Android-пристрої? Що потрібно налаштувати?

Тема 3. Створення гібридних додатків

3.1. Передумови для створення гібридних мобільних застосунків

Гібридні мобільні застосунки створюються з використанням веб-технологій — HTML, CSS, JavaScript/TypeScript — і обгортаються в спеціальний контейнер, що дозволяє їм працювати як повноцінні нативні програми. Щоб реалізувати такі застосунки, розробнику необхідно налаштувати відповідне середовище: встановити Node.js, менеджер пакетів npm, фреймворк (наприклад, Ionic) та інструменти для роботи з мобільними платформами (Android Studio, емулятор AVD).

Важливо розуміти, що створення гібридного додатку — це не просто написання HTML-сторінки. Це повноцінна інженерна робота зі створення архітектури застосунку, управління маршрутизацією, роботою з API, сховищами даних, безпекою та інтерфейсом користувача. Саме завдяки фреймворкам на зразок Ionic розробники можуть створювати швидкі, масштабовані, кросплатформенні рішення з адаптивним дизайном.

3.2. Етапи створення гібридного застосунку

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 21

1. Встановлення середовища розробки

Для розробки необхідно:

- **Node.js** (версія 18 або вище) — платформа виконання JavaScript/TypeScript-проектів;
- **npm (Node Package Manager)** — менеджер залежностей;
- **Ionic CLI** — командний інтерфейс для створення, запуску та білду проектів;
- **Android Studio** — для створення емуляторів і білду APK;
- **Git** — для контролю версій і співпраці.

Встановлення Ionic CLI:

```
npm install -g @ionic/cli
```

Перевірка:

```
ionic -v
```

2. Створення нового проекту

Для створення нового застосунку:

```
ionic start student-card blank --type=angular
```

Тут:

- `student-card` — назва застосунку;
- `blank` — шаблон (порожній стартовий проект);
- `--type=angular` — вибір Angular як фронтенд-фреймворку.

Після створення обов'язково перейти у директорію проекту:

```
cd student-card
```

3. Структура проекту

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 22

У типовому застосунку Ionic + Angular основна структура:

- `src/app/` — основна логіка, сторінки, сервіси;
- `src/assets/` — статичні файли;
- `src/environments/` — налаштування середовища;
- `src/index.html, main.ts` — точка входу;
- `app.module.ts` — конфігурація модулів;
- `app-routing.module.ts` — маршрутизація сторінок.

4. Створення сторінок

Ionic CLI дозволяє створювати сторінки автоматично:

```
ionic generate page about
ionic generate page login
```

Кожна сторінка створюється з чотирьох файлів: `about.page.ts`, `about.page.html`, `about.page.scss`, `about-routing.module.ts`.

5. Запуск у браузері

Для швидкої перевірки функціональності використовується команда:

```
ionic serve
```

Вона компілює додаток і запускає локальний сервер з live reload.

6. Підключення Android-платформи

Після перевірки в браузері додаток можна підготувати до запуску на Android:

1. Додати Capacitor:

```
npm install @capacitor/android
npx cap add android
```

2. Синхронізувати застосунок:

```
ionic build
npx cap sync
```

3. Відкрити в Android Studio:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 23

`npx cap open android`

4. Обрати віртуальний пристрій (AVD) або фізичний телефон, натиснути **Run**.

3.3. Інструменти і технології

Ionic Framework

Ionic надає набір компонентів інтерфейсу, які адаптуються до платформи, і дозволяє створювати застосунки, які виглядають як рідні. Його компоненти (наприклад, `ion-button`, `ion-input`, `ion-card`) мають вбудовану підтримку анімацій, темної теми, адаптації під різні екрани.

Angular

Angular — це фреймворк, що базується на компонентному підході, має потужну систему типізації, підтримку реактивного програмування (RxJS), `dependency injection` і модульність.

Capacitor

Capacitor дозволяє застосунку взаємодіяти з апаратною частиною пристрою. Наприклад, для доступу до камери, геолокації, Bluetooth, локального сховища тощо використовуються плагіни:

```
npm install @capacitor/camera
```

3.4. Основи компонування UI в Ionic

Усі елементи інтерфейсу в Ionic реалізуються через веб-компоненти. Кожен елемент має свій API, події та стилі.

Приклад форми логіну:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 24

```

<ion-content>
  <ion-card>
    <ion-card-header>
      <ion-card-title>Вхід</ion-card-title>
    </ion-card-header>
    <ion-card-content>
      <ion-input      label="Email"
[ (ngModel) ]="email"></ion-input>
      <ion-input      label="Пароль"      type="password"
[ (ngModel) ]="password"></ion-input>
      <ion-button      expand="block"
(click)="login()">Увійти</ion-button>
    </ion-card-content>
  </ion-card>
</ion-content>

```

Такі компоненти автоматично стилізуються залежно від платформи (Android/iOS).

3.5. Практичні рекомендації

1. **Стартуй з простого:** не перевантажуй додаток одразу великою кількістю функцій.
2. **Дотримуйся архітектури:** використовуй сервіси для логіки, моделі для структур даних.
3. **Працюй з Git:** веди контроль версій, роби окремі гілки для функціоналу.
4. **Тестуй одразу:** перевіряй застосунок на Android-емуляторі, фізичному пристрої та у браузері.
5. **Не ігноруй помилки CLI:** уважно читай повідомлення терміналу. Більшість проблем мають прямі рішення.

3.6. Підсумки

Створення гібридного застосунку — це не просто верстка інтерфейсу, а повноцінна розробка з архітектурною побудовою, налаштуванням середовища,

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 25

роботою з платформами Android/iOS, використанням плагінів та оптимізацією роботи застосунку. Завдяки Ionic Framework, Angular і Capacitor сучасні розробники мають у своєму розпорядженні всі необхідні інструменти для створення професійних, масштабованих і продуктивних мобільних рішень.

Тема 4. Інтеграція нативних функцій у гібридних додатках

4.1. Поняття нативних функцій

Нативні функції — це функціональні можливості, які надаються безпосередньо мобільною операційною системою (Android чи iOS) та апаратним забезпеченням пристрою. До таких функцій належать:

- камера, мікрофон;
- акселерометр, гіроскоп;
- Bluetooth, NFC, GPS;
- біометрична автентифікація;
- локальні сповіщення;
- файлові системи;
- мережевий статус та інші.

При розробці нативного застосунку (на Java/Kotlin або Swift) розробник напряму працює з SDK пристрою. У гібридних застосунках — такий доступ реалізується через спеціальні мости (bridges) між WebView та нативним кодом. Раніше цю роль виконував Cordova, сьогодні — більш сучасна платформа Capacitor.

4.2. Що таке Capacitor

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 26

Capacitor — це офіційний нативний рушій від команди Ionic, що дозволяє гібридному додатку взаємодіяти з API мобільного пристрою. Він працює як прошарок між JavaScript-кодом і SDK пристрою.

Особливості Capacitor:

- підтримка Android, iOS, Web з єдиною кодовою базою;
- використання плагінів для підключення нативних функцій;
- можливість написання власних плагінів на Java/Swift;
- підтримка сучасного ES6+ JavaScript/TypeScript;
- автоматичне створення нативних проєктів.

Capacitor має вбудовані плагіни, які охоплюють основні нативні можливості, та відкриту екосистему плагінів від спільноти.

4.3. Підключення плагінів Capacitor

Кожна функція потребує окремого плагіну. Наприклад, для доступу до камери:

```
npm install @capacitor/camera
npx cap sync
```

Використання в Angular-компоненті:

```
import { Camera, CameraResultType } from
 '@capacitor/camera';

async takePhoto() {
  const image = await Camera.getPhoto({
    quality: 90,
    allowEditing: false,
    resultType: CameraResultType.Base64
  });
  this.photo = image.base64String;
}
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 27

Інші приклади плагінів:

- @capacitor/haptics — вібрація;
- @capacitor/geolocation — визначення місцезнаходження;
- @capacitor/filesystem — доступ до локальних файлів;
- @capacitor/network — перевірка наявності з'єднання;
- @capacitor/action-sheet — системне меню вибору.

4.4. Приклад: Камера та галерея

Камера використовується для захоплення зображення або вибору з галереї. Це може бути корисно у додатках з профілем користувача, маркетплейсах, таск-трекерах тощо.

Кроки для інтеграції:

1. Встановити плагін:

```
npm install @capacitor/camera
```

2. Імпортувати:

```
import { Camera, CameraResultType, CameraSource } from
 '@capacitor/camera';
```

3. Виклик функції:

```
const photo = await Camera.getPhoto({
  quality: 70,
  allowEditing: true,
  resultType: CameraResultType.Base64,
  source: CameraSource.Prompt // камера або галерея
});
```

Зображення можна зберегти у Firebase Storage або у локальне сховище (наприклад, Ionic Storage).

4.5. Приклад: Геолокація

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 28

Отримання координат у режимі реального часу використовується в додатках типу «показати завдання поблизу», доставки, мап, безпеки.

Інтеграція:

```
npm install @capacitor/geolocation
```

```
import { Geolocation } from '@capacitor/geolocation';

const coordinates = await
Geolocation.getCurrentPosition();
console.log('Current position:', coordinates);
```

Також можна підписуватись на зміну координат:

```
Geolocation.watchPosition({}, (position) => {
  console.log('Updated:', position);
});
```

4.6. Приклад: Біометрична автентифікація

Підвищує рівень безпеки, зручно для швидкого входу.

Плагіни:

- @aparajita/capacitor-biometric-auth — підтримує FaceID, TouchID.

```
import { NativeBiometric } from
"capacitor-native-biometric";

NativeBiometric.verifyIdentity({
  reason: "Для доступу до ToDo List",
  title: "Автентифікація",
  subtitle: "Будь ласка, підтвердьте вхід"
})
.then(() => this.navigateToTasks())
.catch(() => alert("Вхід скасовано"));
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 29

4.7. Вібрація, offline-режим, Action Sheet

Вібрація:

```
npm install @capacitor/haptics
```

```
import { Haptics, ImpactStyle } from
 '@capacitor/haptics';
Haptics.impact({ style: ImpactStyle.Medium });
```

Offline-перевірка:

```
npm install @capacitor/network
```

```
import { Network } from '@capacitor/network';

const status = await Network.getStatus();
if (!status.connected) {
  alert("Немає інтернету. Застосунок не працює офлайн.");
}
```

Action Sheet (системне меню):

```
npm install @capacitor/action-sheet
```

```
import { ActionSheet } from '@capacitor/action-sheet';

await ActionSheet.showActions({
  title: 'Підтвердіть дію',
  options: [
    { title: 'Видалити', style: 'destructive' },
    { title: 'Скасувати', style: 'cancel' }
  ]
});
```

4.8. Зберігання зображень у Firebase Storage

Після отримання фото з камери, його можна зберегти у Firebase:

1. Ініціалізувати Firebase Storage.
2. Конвертувати фото у формат Blob.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 30

3. Завантажити за допомогою AngularFireStorage:

```
import { AngularFireStorage } from
 '@angular/fire/compat/storage';

const path = `photos/${userId}/${Date.now()}.jpeg`;
const task = this.storage.upload(path,
  base64ToBlob(photoData));
```

Фотографія може зберігатись разом з даними задачі у Firestore як URL.

4.9. Плагіни спільноти

Окрім офіційних плагінів, існує велика кількість плагінів з відкритим кодом:

- голосове введення;
- QR-сканери;
- інтеграція зі сторонніми Bluetooth-пристроями;
- системні діалоги;
- розпізнавання обличчя або тексту.

Усі доступні плагіни розміщено на <https://capacitorjs.com>

4.10. Висновки

Інтеграція нативних функцій — це ключова особливість гібридної розробки, яка дозволяє створювати багатофункціональні додатки з глибоким доступом до апаратного забезпечення. Завдяки Capacitor розробники можуть реалізувати практично будь-який сценарій: фото, геолокація, офлайн-режим, біометрія, сповіщення тощо. Використання стандартних плагінів, сучасної архітектури та Firebase як бекенду дає змогу створювати професійні застосунки без потреби у нативному програмуванні.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 31

ЛАБОРАТОРНА РОБОТА №2

Тема: Створення ToDo-застосунку. Локальне збереження даних із використанням Ionic Storage

Мета роботи: Опанувати створення простого гібридного мобільного додатку з використанням Ionic Framework, реалізувати локальне зберігання інформації за допомогою @ionic/storage-angular, організувати роботу зі списком завдань (створення, редагування, видалення, збереження), ознайомитися з базовими принципами навігації, сервісної логіки та реактивного оновлення інтерфейсу.

Короткі теоретичні відомості:

ToDo-застосунок — один з найпоширеніших прикладів у вивченні розробки. Він дозволяє створювати завдання (tasks), позначати їх як виконані, редагувати, видаляти, переглядати загальний список.

У мобільному середовищі важливо вміти працювати з локальним сховищем. **Ionic Storage** — це абстракція над кількома типами збереження даних (IndexedDB, LocalStorage, SQLite), яка дозволяє:

- ініціалізувати асинхронне сховище;
- зберігати дані у вигляді ключ-значення;
- зчитувати, оновлювати, видаляти дані;
- зберігати масиви та об'єкти через JSON.

Для реалізації ToDo-логіки використовується сервіс (TaskService), який централізує всі CRUD-операції.

Зміст роботи

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 32

Крок 1. Створення нового проєкту:

```
ionic start todo-app blank --type=angular
cd todo-app
```

Крок 2. Додавання Android-платформи (за потреби):

```
npm install @capacitor/android
npm run cap add android
npm run cap sync
```

Крок 3. Встановлення Ionic Storage:

```
npm install @ionic/storage-angular
```

Крок 4. Налаштування Storage у AppModule:

У `app.module.ts` імпортувати модуль:

```
import { IonicStorageModule } from
 '@ionic/storage-angular';
```

```
@NgModule({
  ...
  imports: [
    ...,
    IonicStorageModule.forRoot()
  ],
  ...
})
```

Крок 5. Створення сервісу TaskService:

```
ionic generate service services/task
```

У сервісі:

- створити інтерфейс Task;
- реалізувати методи:
 - `init()` — ініціалізація Storage;
 - `getTasks()`, `addTask()`, `deleteTask()`, `updateTask()`.

Приклад моделі:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 33

```
export interface Task {
  id: string;
  title: string;
  description?: string;
  completed: boolean;
}
```

Крок 6. Створення сторінки Home (список завдань):

- Список завдань (*ngFor);
- Позначення як виконане (checkbox, swipe);
- Видалення (ion-item-sliding, ion-button).

Крок 7. Створення сторінки додавання/редагування завдання:

```
ionic generate page task-details
```

- Поля введення (ion-input, ion-textarea);
- Кнопка «Зберегти»;
- При відкритті з параметром id — редагування;
- Інакше — нове завдання.

Крок 8. Налаштування маршрутизації:

```
У app-routing.module.ts:
```

- додати шляхи /home, /task-details/:id;
- при переході передавати параметри.

Крок 9. Зв'язок між сторінками:

- З Home перейти до TaskDetails по routerLink;
- Після збереження — повернення назад.

Крок 10. Тестування роботи:

- додати, редагувати, видаляти завдання;
- перевірити збереження після перезапуску застосунку;
- перевірити, що список оновлюється автоматично.

Крок 11. Збірка проєкту на Android-пристрій або емулятор:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 34

```
ionic build
npx cap copy
npx cap open android
```

Крок 12. Публікація на git.ztu.edu.ua:

- створити репозиторій;
- зафіксувати зміни:

```
git init
git add .
git commit -m "lab2"
git remote add origin <your-repo-url>
git push -u origin master
```

- додати викладача

Контрольні запитання:

1. Що таке Ionic Storage? Які типи сховищ використовуються під капотом?
2. Як ініціалізується асинхронне сховище в Angular-проєкті?
3. У чому відмінність між LocalStorage та Ionic Storage?
4. Яка роль сервісу у реалізації логіки збереження завдань?
5. Як реалізувати передачу даних між сторінками в Angular?

Тема 5. Інтеграція з Firebase та робота з даними в гібридних додатках

5.1. Що таке Firebase

Firebase — це платформа від Google, яка надає повний набір інструментів для розробки, розгортання, аналітики та підтримки мобільних і веб-застосунків. Вона працює за моделлю **BaaS (Backend as a Service)**, що дозволяє уникнути написання власного серверного коду для базової логіки.

Основні сервіси Firebase, корисні в гібридній мобільній розробці:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 35

- **Firebase Authentication** — автентифікація користувачів (email, Google, Facebook, телефон тощо);
- **Cloud Firestore** — масштабована NoSQL-база даних у реальному часі;
- **Realtime Database** — дерево даних у JSON-форматі;
- **Cloud Storage** — збереження файлів (зображення, документи тощо);
- **Firebase Hosting** — хостинг для веб-застосунків;
- **Firebase Functions** — безсерверна логіка (Cloud Functions for Firebase);
- **Firebase Analytics** — моніторинг поведінки користувачів.

Firebase має офіційний SDK для Angular — **AngularFire**, що забезпечує зручну та реактивну інтеграцію.

5.2. Підключення Firebase до Angular/Ionic-проєкту

Крок 1: Створення проєкту на Firebase

1. Перейдіть на сайт <https://firebase.google.com>.
2. Створіть новий проєкт (наприклад, todo-app-kkn-ldv).
3. У розділі "Authentication" активуйте необхідні методи входу (email/password).
4. У розділі "Firestore Database" створіть базу в режимі тестування.
5. У розділі "Project Settings" → "Web App" створіть застосунок і збережіть конфігураційний об'єкт.

Крок 2: Додавання залежностей

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 36

У терміналі:

```
npm install firebase @angular/fire
```

Крок 3: Налаштування environment.ts

```
export const environment = {
  production: false,
  firebaseConfig: {
    apiKey: '...',
    authDomain: '...',
    projectId: '...',
    storageBucket: '...',
    messagingSenderId: '...',
    appId: '...'
  }
};
```

Крок 4: Імпорт модулів у AppModule

```
import { provideFirebaseApp, initializeApp } from
'@angular/fire/app';

import { provideFirestore, getFirestore } from
'@angular/fire/firestore';

import { provideAuth, getAuth } from
'@angular/fire/auth';

@NgModule({
  imports: [
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 37

```

        provideFirebaseApp(() =>
initializeApp(environment.firebaseConfig)),
        provideFirestore(() => getFirestore()),
        provideAuth(() => getAuth())
    ]
})

```

5.3. Робота з Firestore (Cloud Firestore)

Firestore — це документоорієнтована база даних, у якій:

- дані зберігаються у **колекціях**;
- кожна колекція містить **документи**;
- документи мають унікальні ID і можуть містити вкладені поля/об'єкти.

Приклад колекції `tasks`, де кожен документ — це окреме завдання.

Додавання документа:

```

import { collection, addDoc } from
'firebase/firestore';

const colRef = collection(this.firestore, 'tasks');
await addDoc(colRef, { title: 'Новий таск', done:
false });

```

Отримання даних у реальному часі:

```

import { collectionData } from
'@angular/fire/firestore';

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 38

```

    collectionData(colRef, { idField: 'id'
  })).subscribe(tasks => {
    this.tasks = tasks;
  });

```

Видалення документа:

```

import { deleteDoc, doc } from 'firebase/firestore';
const docRef = doc(this.firestore, 'tasks', taskId);
await deleteDoc(docRef);

```

5.4. Реалізація автентифікації

Firebase дозволяє швидко інтегрувати автентифікацію через email, Google, Facebook, телефон тощо.

Реєстрація користувача:

```

import { createUserWithEmailAndPassword } from
'firebase/auth';
await createUserWithEmailAndPassword(this.auth,
email, password);

```

Вхід:

```

import { signInWithEmailAndPassword } from
'firebase/auth';
await signInWithEmailAndPassword(this.auth, email,
password);

```

Вихід:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 39

```
import { signOut } from 'firebase/auth';
await signOut(this.auth);
```

Отримання поточного користувача:

```
this.auth.onAuthStateChanged(user => {
  if (user) {
    this.userId = user.uid;
  }
});
```

5.5. Захист маршруту (auth-guard)

AngularFire надає механізми для обмеження доступу до маршрутів:

```
import { canActivate, redirectUnauthorizedTo } from
'@angular/fire/auth-guard';
```

```
const redirectToLogin = () =>
redirectUnauthorizedTo(['login']);
```

```
{
  path: 'home',
  component: HomePage,
  ...canActivate(redirectToLogin)
}
```

Також можна реалізувати зворотній редірект: якщо користувач залогінений — переадресувати на головну.

5.6. Прив'язка даних до користувача

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 40

Щоб користувач бачив лише свої дані, у Firestore додається поле `userId`.

При створенні завдання:

```
await addDoc(colRef, {
  title: 'Моє завдання',
  done: false,
  userId: this.userId
});
```

Для отримання даних:

```
import { query, where } from 'firebase/firestore';
const q = query(collection(this.firestore, 'tasks'),
where('userId', '==', this.userId));
```

5.7. Зберігання даних у Storage

Firebase Storage дозволяє зберігати зображення, аудіо, PDF тощо. Для збереження фото:

```
import { AngularFireStorage } from
'@angular/fire/compat/storage';
this.storage.upload('photos/' + name, file);
```

Після цього отримується URL:

```
this.storage.ref('photos/' + name).getDownloadURL()
```

URL може зберігатись у Firestore разом з іншими даними.

5.8. Переваги та обмеження використання Firebase

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 41

Переваги:

- готова інфраструктура;
- швидке розгортання;
- безпека й авторизація «з коробки»;
- масштабування під навантаження;
- реактивність (справжній real-time).

Обмеження:

- обмеження за тарифами (quota);
- деякі типи запитів у Firestore мають обмеження;
- зростання вартості при масштабі;
- складність гнучкої кастомізації без Cloud Functions.

5.9. Висновки

Інтеграція Firebase у гібридні мобільні застосунки на Ionic дає можливість створювати повнофункціональні, безпечні і масштабовані додатки з підтримкою:

- автентифікації,
- роботи з базами даних у реальному часі,
- зберігання файлів,
- аналітики.

Це дозволяє зосередитися на бізнес-логіці і UI/UX, не витрачаючи зусилля на розгортання бекенду. У контексті навчального курсу — це ідеальний бекенд для ToDo-додатків, чатів, профільних застосунків, блогів тощо.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 42

Тема 6. Використання офлайн баз даних у мобільних додатках

6.1. Потреба в офлайн-збереженні

Багато мобільних застосунків повинні працювати навіть за відсутності підключення до інтернету. Це особливо важливо у таких випадках:

- користувач перебуває в зоні зі слабким покриттям або взагалі без зв'язку;
- під час поїздок, у транспорті, в підземних приміщеннях;
- для збереження особистих налаштувань, чернеток, кешованих даних.

Забезпечення офлайн-доступу до даних покращує користувацький досвід, дозволяє уникнути втрати інформації, пришвидшує інтерфейс та зменшує кількість звернень до мережі.

6.2. Види локальних сховищ у мобільних застосунках

У гібридних мобільних додатках доступні кілька типів локального збереження:

1) LocalStorage (Web Storage API)

- Сховище у форматі ключ-значення.
- Доступне у браузері, обмежене ~5MB.
- Синхронне, не придатне для великих об'ємів.
- Застосування: тимчасові налаштування, маркери автентифікації.

2) IndexedDB

- Асинхронна об'єктно-орієнтована база даних у браузері.
- Підтримує запити, індекси, транзакції.
- Придатна для складних офлайн-застосунків.
- Недолік — складність API.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 43

3) SQLite

- Повноцінна реляційна база даних.
- Використовується як нативне сховище через плагіни.
- Ідеальна для офлайн-застосунків зі складними запитами.
- Вимагає знання SQL.

4) Ionic Storage

- Обгортка над вищезазначеними сховищами.
- Універсальний API з fallback-логікою.
- Працює з JSON-об'єктами, простий у використанні.
- Може використовувати SQLite, IndexedDB або LocalStorage в залежності від платформи.

6.3. Переваги Ionic Storage

Ionic Storage — це рекомендований інструмент для локального зберігання у гібридних додатках. Його особливості:

- автоматичне визначення найбільш продуктивного сховища;
- підтримка асинхронної ініціалізації;
- прості методи API: `get()`, `set()`, `remove()`, `clear()`;
- зберігає об'єкти у форматі JSON.

Підключення:

```
npm install @ionic/storage-angular
```

Налаштування:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 44

```
import { IonicStorageModule } from
 '@ionic/storage-angular';
```

```
@NgModule({
  imports: [
    IonicStorageModule.forRoot()
  ]
})
```

Ініціалізація:

```
await this.storage.create()
```

6.4. Приклад: Реалізація ToDo з офлайн-збереженням

1) Модель Task:

```
export interface Task {
  id: string;
  title: string;
  description?: string;
  done: boolean;
}
```

2) Сервіс TaskService:

```
@Injectable({ providedIn: 'root' })
export class TaskService {
  private storageReady = false;
  private tasks: Task[] = [];
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 45

```

constructor(private storage: Storage) {}

async init() {
  await this.storage.create();
  this.storageReady = true;
  const stored = await this.storage.get('tasks');
  this.tasks = stored || [];
}

async getTasks(): Promise<Task[]> {
  return this.tasks;
}

async addTask(task: Task) {
  this.tasks.push(task);
  await this.storage.set('tasks', this.tasks);
}

async deleteTask(id: string) {
  this.tasks = this.tasks.filter(t => t.id !== id);
  await this.storage.set('tasks', this.tasks);
}
}

```

3) Сторінки:

- Home — список завдань.
- TaskDetails — додавання/редагування.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 46

- 4) **Збереження відбувається локально:** усі дії з масивом `tasks` синхронізуються зі сховищем.

6.5. Оновлення UI в реальному часі

Важливо, щоб інтерфейс оновлювався одразу після додавання або видалення завдань. Це можна зробити:

- через `Subject/BehaviorSubject` з `RxJS`;
- через просте оновлення масиву й повторне читання:

```
this.taskService.getTasks().then(tasks => this.tasks = tasks);
```

Також слід

ініціалізувати `TaskService` у `AppComponent` або `HomePage` через `ngOnInit()`.

6.6. Альтернативи: SQLite, NoSQL, Encrypted Storage

SQLite

- Підключається через `@capacitor-community/sqlite`.
- Потрібно створювати таблиці вручну.
- Переваги — складні запити, JOIN-и, фільтри.
- Застосовується в офлайн-бухгалтерії, базах клієнтів тощо.

Encrypted Storage

- Для чутливих даних (токени, паролі).
- Плагіни: `capacitor-secure-storage-plugin`.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 47

Firestore з офлайн-кешуванням

- Firestore підтримує офлайн-синхронізацію.
- Для цього потрібно активувати persistence:

```
import { enableIndexedDbPersistence } from
"firebase/firestore";
enableIndexedDbPersistence(db);
```

6.7. Висновки

Офлайн-збереження — важлива частина UX. У гібридній мобільній розробці це досягається завдяки:

- використанню **Ionic Storage** для простих задач;
- **SQLite** для складних баз даних;
- **Firestore** з включеною офлайн-синхронізацією — як проміжний варіант.

Розробник повинен обрати сховище в залежності від складності проєкту, обсягу даних, необхідності в безпеці й типу операцій (CRUD, фільтрація, сортування тощо).

ЛАБОРАТОРНА РОБОТА №3

Тема: Firebase у мобільних застосунках. Реалізація автентифікації та збереження даних користувача

Мета роботи: Опанувати підключення Firebase до гібридного мобільного застосунку на Ionic + Angular, реалізувати сторінки реєстрації, логіну та збереження даних користувача до хмарної бази Firestore. Також — організувати

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 48

прив'язку завдань до конкретного користувача та обмежити доступ до сторінок залежно від стану автентифікації.

Короткі теоретичні відомості:

Firebase — хмарна BaaS-платформа, яка дозволяє швидко реалізувати серверну частину застосунку. Найчастіше у мобільній розробці використовують:

- **Firebase Authentication** — для входу/реєстрації;
- **Cloud Firestore** — NoSQL база для збереження даних;
- **Firebase Storage** — для зберігання зображень;
- **AngularFire** — бібліотека для зручної інтеграції з Angular.

Прив'язка даних до користувача — критичний етап для багатокористувацьких застосунків. У Firestore документи зберігаються з полем `userId`, яке відповідає `uid` залогіненого користувача. Це дозволяє фільтрувати й відображати лише ті об'єкти, що належать певному користувачу.

Зміст роботи

Крок 1. Реєстрація в Firebase

1. Перейдіть на <https://firebase.google.com>
2. Створіть новий проєкт (наприклад: `todo-app-kkn-ldv`)
3. Додайте вебзастосунок (без хостингу)
4. Збережіть об'єкт конфігурації (`apiKey`, `authDomain`, `projectId` тощо)
5. Увімкніть у розділі **Authentication** методи: Email/Password

Крок 2. Встановлення бібліотек

```
npm install firebase @angular/fire
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 49

Крок 3. Налаштування середовища

У `src/environments/environment.ts` додайте:

```
export const environment = {
  production: false,
  firebaseConfig: {
    apiKey: "...",
    authDomain: "...",
    projectId: "...",
    storageBucket: "...",
    messagingSenderId: "...",
    appId: "..."
  }
};
```

У `app.module.ts`:

```
import { provideFirebaseApp, initializeApp } from
  '@angular/fire/app';
import { provideAuth, getAuth } from
  '@angular/fire/auth';
import { provideFirestore, getFirestore } from
  '@angular/fire/firestore';

@NgModule({
  imports: [
    provideFirebaseApp(() =>
      initializeApp(environment.firebaseConfig)),
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 50

```

        provideAuth(() => getAuth()),
        provideFirestore(() => getFirestore())
    ]
})

```

Крок 4. Реалізація сторінок логіну та реєстрації

```

ionic generate page login
ionic generate page sign-up

```

- У формі реєстрації: email, пароль, підтвердження паролю, ім'я, телефон (за бажанням)
- Кнопки: "Зареєструватися", "Назад"
- У формі логіну: email, пароль, кнопки "Увійти", "Реєстрація"

Крок 5. Створення AuthService

```

ionic generate service services/auth

```

У `auth.service.ts` реалізуйте:

```

import { Auth, signInWithEmailAndPassword,
createUserWithEmailAndPassword, signOut } from
'@angular/fire/auth';

@Injectable({ providedIn: 'root' })
export class AuthService {
    constructor(private auth: Auth) {}

    register(email: string, password: string) {

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 51

```

        return createUserWithEmailAndPassword(this.auth,
email, password);
    }

    login(email: string, password: string) {
        return signInWithEmailAndPassword(this.auth, email,
password);
    }

    logout() {
        return signOut(this.auth);
    }

    getCurrentUser() {
        return this.auth.currentUser;
    }
}

```

Крок 6. Збереження додаткової інформації користувача

У sign-up.page.ts:

- після створення акаунту збережіть ім'я, телефон тощо у Firestore:

```

const userRef = doc(this.firestore,
`users/${user.uid}`);
await setDoc(userRef, {
    email: user.email,
    name: this.name,
    phone: this.phone

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 52

});

Крок 7. Прив'язка завдань до користувача

1. При створенні завдання — додайте поле `userId`:

```
addTask(task: Task) {
  const docRef = addDoc(collection(this.firestore,
    'tasks'), {
    ...task,
    userId: this.userId
  });
}
```

2. При читанні — фільтруйте:

```
const q = query(collection(this.firestore, 'tasks'),
  where('userId', '==', this.userId));
```

Крок 8. Обмеження доступу (auth-guard)

```
npm install @angular/fire/auth-guard
```

У `app-routing.module.ts`:

```
import { canActivate, redirectUnauthorizedTo } from
  '@angular/fire/auth-guard';
```

```
const redirectToLogin = () =>
  redirectUnauthorizedTo(['login']);
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 53

```
{ path: 'home', component: HomePage,
...canActivate(redirectToLogin) }
```

Крок 9. Вхід та вихід із застосунку

- У login.page.ts:

```
this.authService.login(email, password).then(() =>
this.router.navigateByUrl('/home'));
```

- У меню: реалізуйте logout() → navigateByUrl('/login')

Крок 10. Завантаження задач тільки після логіну

- У AppComponent або HomePage, спочатку визначте користувача:

```
onAuthStateChanged(this.auth, user => {
  if (user) {
    this.userId = user.uid;
    this.loadUserTasks();
  }
});
```

Крок 11. Публікація на git.ztu.edu.ua

- Оформіть .gitignore (щоб не пушити node_modules)
- Додайте README
- Додайте викладача до репозиторію

Контрольні запитання:

1. У чому різниця між Firebase Authentication та Firestore?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 54

2. Як можна реалізувати реєстрацію користувача з додатковими полями?
3. Як обмежити доступ до сторінки без авторизації?
4. Як організувати структуру Firestore для багатокористувацького ToDo-додатку?
5. Які потенційні ризики безпеки при збереженні даних у Firestore?

Тема 7. Розробка користувацького інтерфейсу (UI/UX) у мобільних додатках

7.1. Значення UI/UX у мобільній розробці

Успішність мобільного застосунку сьогодні визначається не лише функціональністю, а й зручністю користування. Навіть технічно досконалий застосунок із неінтуїтивним інтерфейсом або повільним відгуком викликає у користувача фрустрацію та знижує шанси на повторне використання.

UI (User Interface) — це зовнішній вигляд додатку: кнопки, поля, шрифти, кольори, розміщення елементів.

UX (User Experience) — це загальний досвід користувача: наскільки швидко він розуміє, як працює додаток, чи логічна навігація, чи все на своєму місці.

У мобільних умовах (маленький екран, використання однією рукою, можливі відволікання) хороше UI/UX значно підвищує ефективність і задоволеність користувача.

7.2. Принципи UI-дизайну в мобільних додатках

1) Мінімалізм

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 55

Чим менше — тим краще. У мобільному інтерфейсі кожен піксель має значення. Елементи не повинні перевантажувати екран, а тільки підкреслювати функціональність.

2) Контраст і читабельність

Високий контраст між фоном і текстом, великі розміри шрифтів (14–16pt для основного тексту), достатні відступи — усе це впливає на сприйняття.

3) Послідовність

Компоненти мають виглядати та поводитися однаково на всіх екранах. Кнопки, кольори, іконки — усе повинно мати єдиний стиль.

4) Адаптивність

Інтерфейс має коректно масштабуватися на різних розмірах екранів: телефони, планшети, foldable-пристрої. Для цього використовуються відсоткові відступи, flex/grid-сітки, адаптивні одиниці (vw, vh, rem).

5) Зрозумілі іконки та підписи

Уникайте незрозумілих значків без підписів. Користувач має одразу розуміти, що означає кожен елемент.

6) Зворотний зв'язок

Після кожної дії (натискання, збереження) має з'являтися індикація: анімація, повідомлення, зміна кольору — це критично для UX.

7.3. Платформні гайдлайни: iOS vs Android

Android

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 56

- Використовує **Material Design**.
- Стандартні елементи: FAB (Floating Action Button), Bottom Navigation, Drawer Menu.
- Основний акцент — на плоских елементах з тінями, чіткі кольори, анімації.

iOS

- Використовує **Human Interface Guidelines**.
- Елементи мають округлі краї, мінімум тіней.
- Часто застосовується таббар внизу, великі заголовки, текстові кнопки.

У **гібридних** додатках фреймворк Ionic автоматично стилізує компоненти залежно від платформи. Наприклад, `ion-button` на iOS матиме вигляд з округлими кутами, а на Android — з плоским стилем Material.

7.4. Компоненти UI в Ionic Framework

Ionic має десятки вбудованих компонентів, адаптованих для мобільного середовища:

- **ion-header, ion-toolbar** — верхні панелі
- **ion-button** — кнопки з різними стилями
(`fill="clear", color="primary"`)
- **ion-card** — інформаційні блоки
- **ion-item, ion-list** — списки
- **ion-input, ion-textarea, ion-select** — форми
- **ion-modal, ion-alert, ion-toast** — діалогові вікна
- **ion-tabs, ion-router-outlet** — навігація

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 57

Усі компоненти мають параметри, які дозволяють кастомізувати вигляд, розміри, кольори, інтеракцію.

7.5. Робота зі стилями (CSS / SCSS)

Кожна сторінка в Ionic має власний SCSS-файл (*.page.scss), у якому можна перевизначати стилі.

Приклад: стилізація кнопки

```
ion-button.custom {
  --background: #007bff;
  --color: white;
  border-radius: 12px;
  padding: 10px;
}
```

Глобальні стилі

У variables.scss можна визначити теми, кольори, шрифти:

```
:root {
  --ion-color-primary: #4a90e2;
  --ion-font-family: "Poppins", sans-serif;
}
```

7.6. UX-архітектура та навігація

Типи навігації:

- **Stack (push/pop)** — переходи між сторінками.
- **Tabs (ion-tabs)** — нижня навігація, зручна для 3–5 основних розділів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 58

- **Side Menu (ion-menu)** — бічне меню для менш частих дій.

Добрі практики:

- уникати вкладеного табів у меню;
- завжди давати можливість повернутися назад (`ion-back-button`);
- відображати активну сторінку (`[routerLinkActive]`).

7.7. Рекомендації з UX

- **Реєстрація/вхід**: уникайте довгих форм, використовуйте автозаповнення.
- **Списки**: додавайте `swipe`-жести для видалення або архівації.
- **Інтеракції**: додайте вібрацію (`Haptics`), підтвердження дій (`Action Sheet`).
- **Помилки**: не змушуйте користувача здогадуватись — виводьте чіткі повідомлення.
- **Темна тема**: додайте перемикач, врахуйте зміну фонів, тіней, контрасту.

7.8. Інструменти дизайну

- **Figma** — найпопулярніший інструмент для прототипування, UI-компонентів, спільної роботи над дизайном.
- **Material Design Kit** — бібліотека компонентів для Android-дизайну.
- **Ionic UI Patterns** — офіційна документація зі зразками.

7.9. Висновки

UI/UX — це не «декорація», а основа ефективного застосунку. Навіть проста ToDo-програма виглядає професійно, якщо її елементи:

- мають чисту структуру,
- адаптуються під платформу,

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 59

- працюють без зависань,
- відповідають очікуванням користувача.

Завдяки **Ionic Framework** та **Angular** розробник може створювати інтерфейси, які виглядають сучасно, реагують миттєво та дотримуються гайдлайнів платформи. Дотримання принципів UI/UX — запорука того, що користувач не лише встановить застосунок, а й продовжить ним користуватися.

Тема 8. Використання Firebase для аутентифікації та моніторингу у мобільних додатках

8.1. Поняття автентифікації

Автентифікація (authentication) — це процес підтвердження особи користувача. Вона є критично важливою у будь-якому застосунку, який:

- зберігає персональні дані;
- дозволяє користувачу виконувати дії, які не можна робити анонімно;
- персоналізує контент або забезпечує збереження прогресу.

Типові варіанти автентифікації в мобільних застосунках:

- електронна пошта + пароль;
- вхід через соцмережі (Google, Facebook, Apple);
- телефон (OTP);
- біометрія (Touch ID, Face ID).

Firebase надає вбудований сервіс **Firebase Authentication**, який спрощує реалізацію будь-якого з цих методів з високим рівнем безпеки, мінімальним налаштуванням і хорошою документацією.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 60

8.2. Firebase Authentication

Firebase Authentication — це сервіс для реєстрації та входу користувачів. Він підтримує:

- email + пароль;
- Google, Facebook, GitHub, Microsoft;
- телефонна автентифікація з OTP;
- анонімний вхід.

Firebase бере на себе:

- безпечне зберігання паролів;
- керування сесіями;
- захист від атак;
- перевірку токенів.

Усі користувачі зберігаються в консолі Firebase у розділі **Authentication** → **Users**.

8.3. Реалізація email-пароль автентифікації

1) Створення акаунта (реєстрація):

```
import { createUserWithEmailAndPassword } from
'firebase/auth';
await createUserWithEmailAndPassword(auth, email,
password);
```

2) Вхід користувача:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 61

```
import { signInWithEmailAndPassword } from
'firebase/auth';
await signInWithEmailAndPassword(auth, email, password);
```

3) Вихід:

```
import { signOut } from 'firebase/auth';
await signOut(auth);
```

4) Відстеження стану автентифікації:

```
onAuthStateChanged(auth, user => {
  if (user) {
    this.userId = user.uid;
  } else {
    this.router.navigateByUrl('/login');
  }
});
```

8.4. Безпечне збереження додаткових даних користувача

Хоча Firebase зберігає ім'я та email, часто потрібна додаткова інформація — телефон, дата народження, фото тощо. Вона зберігається окремо у Firestore:

```
const userDoc = doc(this.firestore,
`users/${user.uid}`);
await setDoc(userDoc, {
  name: this.name,
  phone: this.phone,
  role: 'user'
});
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 62

Такий підхід дозволяє створити власну модель користувача з гнучкою структурою.

8.5. Обмеження доступу до сторінок (auth guard)

AngularFire має вбудовані механізми для захисту маршрутів. Це дозволяє:

- перенаправити неавторизованого користувача на сторінку логіну;
- запобігти входу на сторінку логіну, якщо користувач уже авторизований.

```
import { canActivate, redirectUnauthorizedTo,
  redirectLoggedInTo } from '@angular/fire/auth-guard';
```

```
const toLogin = () => redirectUnauthorizedTo(['login']);
const toHome = () => redirectLoggedInTo(['home']);
```

```
{ path: 'home', component: HomePage,
  ...canActivate(toLogin) }
{ path: 'login', component: LoginPage,
  ...canActivate(toHome) }
```

8.6. Аутентифікація з використанням Google

Google Login — популярна опція для користувачів. Щоб її активувати:

1. Додати у Firebase → Authentication → Sign-in method → Google.
2. Встановити плагін Google Plus (на Android) або використовувати Firebase Web SDK для веб/гібридного застосунку.

Вхід виглядає так:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 63

```
import { GoogleAuthProvider, signInWithPopup } from
'firebase/auth';
```

```
const provider = new GoogleAuthProvider();
await signInWithPopup(this.auth, provider);
```

У гібридному застосунку можна використовувати WebView-вхід або плагіни Capacitor.

8.7. Реалізація "Забув пароль"

Firebase дозволяє надсилати листа з посиланням на скидання пароля:

```
import { sendPasswordResetEmail } from 'firebase/auth';

await sendPasswordResetEmail(this.auth, email);
```

Це додає професійності застосунку і дозволяє уникнути втрати облікового запису.

8.8. Аналітика з Firebase Analytics

Firebase Analytics — сервіс збору статистики про використання додатку:

- кількість користувачів, сесій;
- найпопулярніші екрани;
- події: натискання кнопок, вхід, реєстрація, покупки;
- геолокація, пристрої, версії ОС.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 64

У Web/Hybrid-додатках аналітику можна підключити через `gtag.js`, або через Google Tag Manager. У нативних застосунках (через Capacitor або Android/iOS) — через SDK.

Подія:

```
gtag('event', 'login', {
  method: 'email'
});
```

Важливі показники:

- DAU (daily active users),
- Retention (утримання користувачів),
- Funnel (послідовність дій).

8.9. Висновки

Firebase надає готову, безпечну та гнучку систему автентифікації для мобільних застосунків. Основні переваги:

- підтримка різних типів входу;
- простий і зрозумілий API;
- інтеграція з Firestore для розширення моделі користувача;
- можливість реалізувати повний авторизаційний цикл (реєстрація, логін, logout, забув пароль);
- захист маршрутів і моніторинг активності.

Крім того, Firebase Analytics дає змогу оцінювати реальну поведінку користувачів і вдосконалювати застосунок відповідно до статистики.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 65

ЛАБОРАТОРНА РОБОТА №4

Тема: Використання нативних функцій у мобільних застосунках. Збереження файлів у хмарі

Мета роботи: Опанувати доступ до нативних функцій пристрою (камера, вібрація, біометрія, дата/час, Action Sheet), навчитись зберігати зображення у Firebase Storage, розширити модель завдання додатковими параметрами. Також — обмежити доступ до додатку у разі відсутності інтернету.

Короткі теоретичні відомості:

Capacitor — офіційна платформа від Ionic, яка забезпечує міст між вебкодом і нативним API пристрою. Вона дозволяє використовувати камеру, GPS, біометрію, файлову систему тощо через стандартні плагіни.

Firebase Storage — сервіс для збереження файлів у хмарі (зображень, документів тощо). Після завантаження файл отримує унікальний URL, який можна зберегти у базу даних (Firestore).

Firestore — NoSQL база, яка підтримує зберігання складних структур JSON-документів із будь-якими додатковими полями.

Зміст:

Крок 1. Розширення моделі Task

Модифікуйте інтерфейс:

```
export interface Task {
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 66

```

    id: string;
    title: string;
    description?: string;
    dueDate?: string;
    photoUrl?: string;
    isDone: boolean;
  }

```

Крок 2. Додавання поля дати/часу

Встановіть плагін (за бажанням):

```
npm install @capacitor-community/date-picker
```

Або використайте ion-datetime:

```

<ion-item>
  <ion-label position="floating">Дата</ion-label>
  <ion-datetime display-format="DD.MM.YYYY"
    [(ngModel)]="task.dueDate"></ion-datetime>
</ion-item>

```

Крок 3. Додавання фотографії до завдання

```
npm install @capacitor/camera
```

У компоненті:

```

import { Camera, CameraResultType, CameraSource } from
  '@capacitor/camera';

async selectPhoto() {

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 67

```

const image = await Camera.getPhoto({
  quality: 70,
  resultType: CameraResultType.Base64,
  source: CameraSource.Prompt
});
this.task.photoData = image.base64String;
}

```

Крок 4. Завантаження фото у Firebase Storage

```
npm install @angular/fire
```

У task.service.ts:

```

import { AngularFireStorage } from
 '@angular/fire/compat/storage';
import { finalize } from 'rxjs/operators';

uploadImage(task: Task, base64: string): Promise<string>
{
  const path = `tasks/${Date.now()}.jpg`;
  const ref = this.storage.ref(path);
  const task$ = ref.putString(base64, 'base64', {
contentType: 'image/jpeg' });

  return new Promise((resolve) => {
    task$.snapshotChanges().pipe(
      finalize(() => {
        ref.getDownloadURL().subscribe(url =>
resolve(url));

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 68

```

    })
    ).subscribe();
  });
}

```

Збережіть `photoUrl` у `Firestore` разом із рештою полів.

Крок 5. Додавання вібрації при створенні завдання

```

npm install @capacitor/haptics
import { Haptics, ImpactStyle } from
 '@capacitor/haptics';
Haptics.impact({ style: ImpactStyle.Medium });

```

Крок 6. Додавання підтвердження на видалення (Action Sheet)

```

npm install @capacitor/action-sheet
import { ActionSheet } from '@capacitor/action-sheet';

async confirmDelete() {
  const result = await ActionSheet.showActions({
    title: 'Видалити завдання?',
    options: [
      { title: 'Так', style: 'destructive' },
      { title: 'Скасувати', style: 'cancel' }
    ]
  });

  if (result.index === 0) this.deleteTask();
}

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 69

}

Крок 7. Біометрична автентифікація (Face ID / Touch ID)

```
npm install capacitor-native-biometric
import { NativeBiometric } from
'capacitor-native-biometric';
```

```
NativeBiometric.verifyIdentity({
  reason: "Доступ до завдань",
  title: "Вхід"
}).then(() => {
  this.router.navigateByUrl('/home');
}).catch(() => {
  alert('Доступ скасовано');
});
```

Крок 8. Блокування додатку без інтернету

```
npm install @capacitor/network
import { Network } from '@capacitor/network';

const status = await Network.getStatus();
if (!status.connected) {
  alert("Додаток недоступний без інтернету.");
  this.router.navigateByUrl('/no-internet');
}
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 70

Контрольні запитання:

1. Як працює плагін Capacitor Camera і чим відрізняються типи `CameraResultType`?
2. У чому різниця між локальним і хмарним зберіганням зображень?
3. Як забезпечити підтвердження дії через Action Sheet?
4. Які переваги біометричної автентифікації перед звичайним логіном?
5. Як визначити стан підключення до мережі в мобільному застосунку?

Тема 9. Розробка користувацького інтерфейсу у гібридних мобільних додатках

9.1. Значення якісного інтерфейсу

Користувацький інтерфейс (UI) — це візуальна оболонка застосунку, через яку користувач взаємодіє з функціоналом. У мобільному середовищі інтерфейс повинен бути:

- простим і зрозумілим;
- адаптивним до різних розмірів екранів;
- швидким у відгуку;
- візуально привабливим.

Гарний інтерфейс — це не лише естетика, а й підвищення ефективності, довіри та лояльності користувача.

9.2. Принципи мобільного UI-дизайну

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 71

Адаптивність

Інтерфейс має коректно масштабуватись від маленьких екранів телефонів до великих планшетів. Застосовується flexbox, відсоткові ширини, медіа-запити.

Однорідність

Користувач очікує послідовності. Кнопки мають однаковий стиль і поведінку. Заголовки — єдиний розмір, відступи — стабільні.

Зручність для пальців

Елементи мають бути достатньо великими (мінімум 44×44 px). Уникайте дрібних іконок, щільних списків.

Обмежений простір

В одному екрані бажано не перевантажувати користувача. Використовуйте гест-інтерфейс, розкриті меню, вкладки.

Системні кольори й теми

Кольори повинні бути читабельними, з хорошим контрастом. Підтримка темної теми — стандарт сучасної мобільної розробки.

9.3. Основні UI-компоненти Ionic

ion-header, ion-toolbar, ion-title — верхня панель з навігаційними елементами.

ion-content — основний простір сторінки, підтримує скрол.

ion-button — кнопка з адаптацією під платформу (Android/iOS).

ion-input — поле вводу (email, текст, пароль).

ion-list / ion-item — списки, які можуть містити тайтл, опис, іконку, слайд-жести.

ion-card — інформаційні блоки із заголовком, зображенням, діями.

ion-modal / ion-alert / ion-toast — модальні повідомлення, індикатори дій.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 72

Ці компоненти легко кастомізуються через змінні стилів або SCSS.

9.4. Побудова сіток та адаптивних макетів

Flexbox

Типовий механізм для побудови сітки:

```
<ion-grid>
  <ion-row>
    <ion-col size="6">Ліво</ion-col>
    <ion-col size="6">Право</ion-col>
  </ion-row>
</ion-grid>
```

Stack layout

Для вертикального компоновання:

```
<div class="stack">
  <h1>Заголовок</h1>
  <ion-button>Кнопка</ion-button>
</div>

.stack {
  display: flex;
  flex-direction: column;
  gap: 16px;
}
```

Responsive utility

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 73

Іonic підтримує класи адаптивності (ion-hide, ion-margin, ion-padding) для швидкої побудови респонсив-інтерфейсу.

9.5. Робота з темами та стилями

У `variables.scss` можна задавати кастомну кольорову схему:

```
:root {
  --ion-color-primary: #3f51b5;
  --ion-color-secondary: #ff9800;
}
```

Темна тема:

```
@media (prefers-color-scheme: dark) {
  :root {
    --ion-background-color: #121212;
    --ion-text-color: #ffffff;
  }
}
```

Користувач може перемикатись між темами, а розробник — динамічно застосовувати стилі через `document.body.classList`.

9.6. Анімації, переходи та інтерфейсні ефекти

- **ion-router-outlet** автоматично додає анімації при переході між сторінками.
- **ion-loading**, **ion-toast**, **ion-progress-bar** — візуальний зворотний зв'язок.
- Кастомні анімації можна реалізувати через CSS або `AnimationController`.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 74

Наприклад:

```
const animation = this.animationCtrl.create()
  .addElement(this.el.nativeElement)
  .duration(500)
  .fromTo('opacity', '0', '1');
animation.play();
```

9.7. Управління станом інтерфейсу

Angular підтримує:

- реактивні форми (FormGroup, FormControl);
- ngIf, ngFor, async pipes;
- RxJS для реактивного керування станом.

Приклад відображення списку завдань:

```
<ion-list>
  <ion-item *ngFor="let task of tasks$ | async">
    {{ task.title }}
  </ion-item>
</ion-list>
```

9.8. Навігація та маршрутизація

Використовується RouterModule:

```
{ path: 'home', component: HomePage },
{ path: 'task/:id', component: TaskDetailsPage }
```

Навігація у коді:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 75

```
this.router.navigate(['/task', task.id]);
```

Інтерфейсні патерни:

- таби (ion-tabs);
- drawer menu (ion-menu);
- back button (ion-back-button).

9.9. Підтримка декількох мов (інтернаціоналізація)

Підключення @ngx-translate/core дозволяє додати підтримку кількох мов, перемикач, автоматичне визначення мови системи.

```
npm install @ngx-translate/core
@ngx-translate/http-loader
```

9.10. Висновки

Розробка UI у мобільному застосунку — це не лише додавання кнопок та текстів, а цілісний підхід до зручності, логіки розміщення, адаптивності, зворотного зв'язку. Завдяки компонентам Ionic, темам, CSS-змінним та гнучкій архітектурі Angular, можна створити сучасний, зручний і візуально приємний інтерфейс з мінімальними зусиллями.

Тема 10. Просунутий мобільний UI/UX-дизайн

10.1. Вступ

Базовий інтерфейс мобільного застосунку може забезпечити функціональність, але саме **просунутий дизайн** відрізняє професійний продукт від аматорського. Це включає:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 76

- сучасні візуальні тренди;
- мікроанімації та інтерфейсні ефекти;
- адаптацію під темну тему;
- глибоку інтеграцію з UX-дослідженнями.

Ця тема присвячена глибшому розумінню того, як створювати привабливі, інтуїтивні, динамічні інтерфейси, які викликають позитивні емоції у користувача.

10.2. Тренди мобільного UI у 2020-х роках

Material You (Google)

- Динамічна адаптація кольорів до шпалер;
- М'які тіні, великі відступи, округлі кути;
- Інтуїтивна ієрархія елементів.

Neumorphism

- Імітація м'яких світлотіней;
- Об'ємні кнопки;
- Використовується для мінімалістичних застосунків.

Glassmorphism

- Прозорість, ефект скла;
- Контрастний фон і розмиття;
- Використовується у віджетах, картках, модалках.

Dark Mode by Default

- Темна тема не лише як опція, а як базовий стиль;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 77

- Зменшення навантаження на очі;
- Покращена автономність на OLED-екранах.

10.3. Анімації та мікровзаємодії

Мікроанімації — це короткі, легкі рухи, які супроводжують дії користувача:

- натискання кнопки → зміна кольору;
- додавання в кошик → елемент “літає” в іконку;
- помилки → легке тремтіння поля.

Вони:

- покращують зворотний зв’язок;
- роблять застосунок “живим”;
- скорочують когнітивне навантаження.

Іоніс підтримує:

- нативні анімації через `AnimationController`;
- кастомні CSS-анімації (`transition`, `keyframes`);
- інтеграцію з `Angular Animations`.

10.4. Візуальна ієрархія

Користувач сканує екран, а не читає його. Щоб керувати його увагою:

- **заголовок** — найбільший і жирний;
- **акцентні кнопки** — яскраві, з великими зонами натискання;
- **поля** — однакової висоти, з чітким відступом;
- **вторинні дії** — сірі, менш помітні.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 78

Золотий принцип: один основний call-to-action на екран.

10.5. Робота з типографікою

- Основний шрифт: 14–16 pt;
- Заголовки: 20–24 pt;
- Інтерліньяж: 1.5–1.8;
- Не більше 2 шрифтів у застосунку;
- Важливість дотримання **адаптивної типографіки**:

```
h1 {
  font-size: clamp(1.5rem, 4vw, 2rem);
}
```

10.6. Колірні схеми

Основні принципи:

- Використовуйте палітру з 3–5 кольорів: Primary, Secondary, Background, Surface, Error.
- Контраст для тексту $\geq 4.5:1$.
- Primary → дії; Secondary → акценти; Surface → фони.

Інструменти:

- <https://material.io/resources/color/>
- <https://coolors.co/>

10.7. Адаптація під темну тему

Темна тема — стандарт в Android та iOS. Ключові аспекти:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 79

- світлий текст на темному фоні;
- уникайте чисто чорного — використовуйте #121212, #1C1C1E;
- перевірка в режимі prefers-color-scheme: dark.

```
@media (prefers-color-scheme: dark) {
  ion-content {
    --background: #121212;
    --color: #fff;
  }
}
```

10.8. UX-дослідження

Перед впровадженням інтерфейсу проводиться:

- **аналіз потреб користувача** (interviews, опитування);
- **створення прототипів** (Wireframe → Low-fidelity → High-fidelity);
- **тестування**: спостереження за тим, як користувач виконує типові дії.

Це дозволяє уникнути помилок і створити інтерфейс, що відповідає реальним очікуванням.

10.9. Прототипування в Figma

Figma — найпопулярніший інструмент для UI/UX-дизайну:

- створення компонентів, автошарів, адаптивних фреймів;
- інтерактивні прототипи з переходами;
- підтримка спільної роботи над макетами;
- інтеграція з Tailwind, Ionic, Flutter через плагіни.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 80

10.10. Приклади UI/UX-антипатернів

- Декілька однакових кнопок на сторінці → конфлікт СТА.
- Інтерактивні елементи без візуального зворотного зв'язку.
- Неконтрастний текст на фоновому зображенні.
- Недоступність (відсутність фокус-станів, aria-label тощо).

10.11. Висновки

Просунутий UI/UX — це:

- не просто кольори й іконки, а чітка структура, анімація, досвід;
- дизайн, що адаптується до користувача та контексту;
- візуальна логіка, яка скорочує шлях до цілі.

Гарний UI/UX — це коли користувач **забуває, що він у застосунку**, бо все працює інтуїтивно. Це одна з головних переваг, яку гібридний застосунок може реалізувати нарівні з нативним, завдяки сучасним фреймворкам.

Тема 11. Оптимізація продуктивності гібридних мобільних додатків

11.1. Вступ

Продуктивність мобільного застосунку — це не лише швидкість його роботи, а й відчуття плавності, швидкого відгуку, безперервності. Особливо це критично для гібридних додатків, де HTML/CSS/JS-код виконується у WebView. Оптимізація продуктивності дозволяє досягти:

- меншого часу завантаження;
- плавності анімацій;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 81

- зменшення споживання енергії;
- відсутності лагів при взаємодії.

У цій темі розглянемо технічні підходи та практичні рекомендації щодо підвищення ефективності гібридних застосунків на Ionic + Angular.

11.2. Причини повільної роботи гібридного застосунку

- Занадто велика початкова збірка;
- Надмірне використання сторонніх бібліотек;
- Часті DOM-оновлення;
- Неоптимізовані запити до серверу;
- Неочищені таймери, підписки;
- Важкі або необроблені зображення;
- Неправильне управління життєвим циклом компонентів.

11.3. Поради щодо підвищення продуктивності

1. Lazy loading модулів

Angular дозволяє відкладено завантажувати модулі, які не потрібні на старті:

```
{      path:      'settings',      loadChildren:      ()      =>
import('./pages/settings/settings.module').then(m      =>
m.SettingsModule) }
```

Це зменшує об'єм початкового бандлу і прискорює запуск.

2. *Використання *trackBy* у *ngFor*

```
<ion-item      *ngFor="let      item      of      tasks;      trackBy:
trackByFn">
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 82

```
trackByFn(index, item) {
  return item.id;
}
```

Це запобігає перерендеру всього списку при зміні лише одного елемента.

3. Зменшення залежностей

Не додавайте бібліотек “про всяк випадок”. Кожна зайва залежність:

- збільшує розмір проєкту;
- уповільнює збірку;
- потенційно тягне за собою сторонні залежності.

11.4. Оптимізація ресурсів

Зображення

- Використовуйте формати WebP або AVIF.
- Зменшуйте роздільну здатність під мобільні екрани.
- Завантажуйте зображення лише за потреби (lazy images).
- У Firebase Storage: зберігайте копії для mobile/HD.

Іконки

Використовуйте векторні (SVG), а не PNG. Уникайте надлишкових іконок з icon-pack’ів.

Шрифти

- Мінімізуйте кількість шрифтів та їх ваг (bold, regular);
- Використовуйте system fonts, коли це можливо;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 83

- Уникайте завантаження Google Fonts через зовнішні посилання — інкапсулюйте у локальні файли.

11.5. Мемоізація, `debounce`, `throttle`

При роботі з користувацькими подіями або запитами до сервера важливо не виконувати їх надто часто:

```
@HostListener('scroll', ['$event'])
debouncedScroll = debounce((event) => {
  // обробка скролу
}, 300);
```

Для `debounce/throttle` можна використовувати:

- `lodash.debounce`
- `rxjs/operators/debounceTime`

11.6. Service Workers і кешування

Angular підтримує **PWA**-режим. За допомогою Service Workers можна:

- кешувати ресурси;
- зберігати офлайн-дані;
- зменшити кількість запитів.

```
ng add @angular/pwa
```

Це значно покращує продуктивність при повторних відвідуваннях.

11.7. Інструменти для моніторингу

Lighthouse (Chrome DevTools)

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 84

- Аналіз продуктивності, доступності, SEO;
- Показує час завантаження, виявляє блокуючі ресурси.

Firestore Performance Monitoring

- Вимірює реальний час виконання транзакцій;
- Показує повільні запити, затримки завантаження екрану;
- Дозволяє додавати кастомні мітки у код:

```
import { trace } from '@firebase/performance';
const traceLogin = trace(performance, 'login_flow');
traceLogin.start();
// ...
traceLogin.stop();
```

Angular Profiler

- Використовується для виявлення надмірного ререндеру;
- Доступний через Augury (розширення Chrome).

11.8. Налаштування білду (production mode)

У продакшн-збірці:

```
ionic build --prod
```

Відбувається:

- мінімізація CSS/JS;
- tree-shaking (видалення непотрібного коду);
- відключення Angular dev tools.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 85

11.9. Уникайте “важких” шаблонів

- Не використовуйте багато `ngIf/ngFor` у одному шаблоні;
- Уникайте вкладених компонентів без потреби;
- Слідкуйте за об’ємом DOM.

Краще поділити великі шаблони на менші компоненти та використовувати `ChangeDetectionStrategy.OnPush`.

11.10. Висновки

Оптимізація продуктивності — це постійний процес. На відміну від вебу, мобільний користувач швидко покидає повільний застосунок. Саме тому важливо:

- використовувати `lazy loading`;
- зменшувати бандл;
- оптимізувати запити, зображення, структуру шаблонів;
- моніторити реальну продуктивність за допомогою Lighthouse, Firebase, профайлерів.

Гібридний застосунок може працювати **так само плавно, як і нативний**, якщо дотримуватись сучасних практик оптимізації.

Тема 12. Адаптація мобільного інтерфейсу під різні платформи та типи екранів

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 86

12.1. Вступ

Мобільні застосунки використовуються на широкому спектрі пристроїв — від компактних смартфонів до планшетів і foldable-девайсів. Крім того, вони працюють у двох основних екосистемах — **Android** і **iOS**, кожна з яких має власні дизайнерські гайдлайни, стилі взаємодії та поведінку системних елементів.

Адаптація застосунку під платформу та екран — це не лише технічне питання, а частина UX-дизайну, що безпосередньо впливає на задоволеність користувача.

12.2. Основні виклики при адаптації

- Різні розміри екранів (від 320×568 до 1440×2960 і більше);
- Пропорції екрана (4:3, 16:9, 19.5:9, 21:9);
- Наявність/відсутність фізичних кнопок;
- Вирізи (notches), закруглені краї;
- Поведінка клавіатури;
- Особливості платформних гайдлайнів.

12.3. Підходи до адаптивності

1) Responsive Layout

Застосовується для адаптації до розміру вікна:

- використання відносних одиниць (% , vw, vh);
- побудова інтерфейсу через `ion-grid`, `flex`, `CSS grid`;
- приховування/відображення елементів через `media queries`.

2) Platform-based UI

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 87

Ionic автоматично адаптує компоненти під платформу:

- `ion-button` — має різний вигляд у iOS і Android;
- `ion-back-button` — змінює іконку та позицію;
- `ion-mode="ios"` або `"md"` — дозволяє явно задати стиль.

3) Adaptive UX

Залежно від платформи, можна змінювати логіку:

- у iOS більше використовується таббар знизу;
- в Android — drawer menu зліва.

У коді можна визначити платформу:

```
import { Platform } from '@ionic/angular';

constructor(private platform: Platform) {
  if (this.platform.is('ios')) {
    // iOS специфічна поведінка
  }
}
```

12.4. Робота з адаптивними компонентами Ionic

ion-grid — потужна сітка з адаптивною розміткою:

```
<ion-grid>
  <ion-row>
    <ion-col size="12" size-md="6">Колонка</ion-col>
    <ion-col size="12" size-md="6">Колонка</ion-col>
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 88

```
</ion-row>
</ion-grid>
```

ion-split-pane — дозволяє створити дві панелі на великих екранах (наприклад, меню зліва + контент справа):

```
<ion-split-pane contentId="main">
  <ion-menu contentId="main">...</ion-menu>
  <ion-router-outlet id="main"></ion-router-outlet>
</ion-split-pane>
```

12.5. Масштабування шрифтів та елементів

- Не використовуйте фіксовані розміри (px), натомість — em, rem, vw.
- Для заголовків — clamp():

```
font-size: clamp(1.2rem, 2.5vw, 2rem);
```

- Використовуйте ion-text, ion-label замість p, щоб підтримувати стилі фреймворку.

12.6. Управління станами екрана

Зміна орієнтації:

```
screen.orientation.onChange = () => {
  console.log('Орієнтація змінилася');
};
```

Виявлення розміру вікна:

```
window.innerWidth < 600 ? 'мобільний' : 'планшет';
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 89

Можна створити власні сервіси адаптації (наприклад, BreakpointService), які зберігають поточний стан інтерфейсу.

12.7. Робота з Keyboard, Safe Area, Insets

Клавіатура

У iOS та Android клавіатура може закривати поля:

- Використовуйте `ion-content scroll="true"` для автоскролу;
- Додайте відступи, щоб уникнути перекриття.

Insets / Safe Areas

Використовуйте:

```
padding-top: env(safe-area-inset-top);
```

Щоб забезпечити коректне відображення на екранах з notch'ами або системними панелями.

12.8. Підтримка планшетів та foldable-пристроїв

- Використовуйте `ion-split-pane` для багатостолбцевих макетів;
- Реалізуйте окремі компонування: одна сторінка — список, інша — деталі;
- Додавайте адаптивні класи: `tablet-only`, `mobile-only`.

12.9. Тестування інтерфейсу

- Chrome DevTools → Device Toolbar;
- Android Studio → емулятори різних пристроїв;
- iOS Simulator;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 90

- <https://screensizemap.com> — для перевірки адаптивності;
- Інструменти з Lighthouse, Percy, Appltools — для візуального тестування.

12.10. Висновки

Сучасний мобільний застосунок повинен бути:

- **адаптивним** — під різні екрани;
- **реактивним** — під різні події (клавіатура, зміна орієнтації);
- **адаптованим** — під очікування користувача з iOS або Android.

Завдяки можливостям Ionic та Angular, більшість цих функцій реалізується швидко та ефективно. Однак саме увага до деталей і тестування під різні сценарії роблять інтерфейс універсальним і зручним.

ЛАБОРАТОРНА РОБОТА №5

Тема: Локалізація, геолокація, мапа та push-нотифікації в гібридних мобільних додатках

Мета роботи: Навчитися реалізовувати багатомовний інтерфейс (локалізацію), визначати місцезнаходження користувача, відображати геолокаційні дані на інтерактивній карті (OpenStreetMap або Google Maps), надсилати push-сповіщення через Firebase Cloud Messaging.

Короткі теоретичні відомості:

- **Локалізація** дозволяє перекладати інтерфейс додатку кількома мовами, визначати мову системи або дозволяти перемикання вручну.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 91

- **Геолокація** — це отримання координат поточного розташування користувача через GPS, Wi-Fi або мережу.
- **OpenStreetMap (OSM)** — вільна мапа з відкритим API.
- **Leaflet.js** — JavaScript-бібліотека для роботи з OSM.
- **Firebase Cloud Messaging (FCM)** — сервіс Google для надсилання push-нотифікацій на мобільні пристрої.

Хід виконання лабораторної роботи:

КРОК 1. Реалізація локалізації інтерфейсу

Встановлення Translate:

```
npm install @ngx-translate/core
@ngx-translate/http-loader
```

Налаштування:

У `app.module.ts`:

```
import { TranslateModule, TranslateLoader } from
'@ngx-translate/core';
import { HttpClientModule, HttpClient } from
'@angular/common/http';
import { TranslateHttpLoader } from
'@ngx-translate/http-loader';

export function HttpLoaderFactory(http: HttpClient) {
  return new TranslateHttpLoader(http, './assets/i18n/',
'./json');
}

@NgModule({
  imports: [
    HttpClientModule,
    TranslateModule.forRoot({
      loader: {
        provide: TranslateLoader,
        useFactory: HttpLoaderFactory,
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 92

```

        deps: [HttpClient]
      }
    })
  ]
})

```

Створіть переклади:

- assets/i18n/uk.json
- assets/i18n/en.json

```

// uk.json
{
  "TASKS": "Завдання",
  "ADD_TASK": "Додати завдання",
  "WELCOME": "Ласкаво просимо"
}

```

У шаблоні:

```
<ion-title>{{ 'TASKS' | translate }}</ion-title>
```

Перемикання мов:

```
this.translate.use('uk');
```

КРОК 2. Отримання геолокації

Встановлення плагіну:

```
npm install @capacitor/geolocation
```

Використання:

```

import { Geolocation } from '@capacitor/geolocation';

const coordinates = await
  Geolocation.getCurrentPosition();
this.latitude = coordinates.coords.latitude;
this.longitude = coordinates.coords.longitude;

```

КРОК 3. Відображення карти OpenStreetMap (Leaflet)

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 93

Встановлення Leaflet:

```
npm install leaflet
npm install --save-dev @types/leaflet
```

Стилі у angular.json:

```
"styles": [
  "node_modules/leaflet/dist/leaflet.css"
]
```

Ініціалізація карти:

```
import * as L from 'leaflet';

map: any;

ngAfterViewInit() {
  this.map = L.map('map').setView([this.latitude,
    this.longitude], 13);

  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '© OpenStreetMap contributors'
  }).addTo(this.map);

  L.marker([this.latitude,
    this.longitude]).addTo(this.map);
}
```

У HTML:

```
<div id="map" style="height: 300px;"></div>
```

КРОК 4. Push-нотифікації через Firebase

У Firebase:

1. Увімкніть **Cloud Messaging**.
2. Згенеруйте ключ VAPID (якщо Web), або **server key** для Android/iOS.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 94

У проєкті:

```
npm install @capacitor/push-notifications
```

Ініціалізація:

```
import { PushNotifications } from
 '@capacitor/push-notifications';

PushNotifications.requestPermissions().then(result => {
  if (result.receive === 'granted') {
    PushNotifications.register();
  }
});

PushNotifications.addListener('registration', token => {
  console.log('Push token:', token.value);
});

PushNotifications.addListener('pushNotificationReceived'
, notification => {
  alert(notification.title + '\n' + notification.body);
});
```

Надсилання:

- Через Firebase Console (Cloud Messaging → New Notification);
- Через REST API / Node.js / Postman.

Контрольні запитання:

1. Як працює механізм локалізації в Angular?
2. Чим відрізняються Google Maps та OpenStreetMap з точки зору використання?
3. Як отримати координати користувача на мобільному пристрої?
4. Які кроки необхідні для надсилання push-повідомлень?
5. Які переваги має Leaflet у гібридних додатках?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 95

Тема 13. Тестування гібридних мобільних додатків

13.1. Вступ

Тестування мобільного застосунку — це невід’ємна частина життєвого циклу програмного забезпечення. Воно дозволяє:

- забезпечити стабільність роботи;
- знайти й виправити помилки;
- покращити користувацький досвід;
- перевірити сумісність із різними пристроями та платформами.

У гібридних застосунках (Ionic + Angular) застосовуються ті ж типи тестування, що й у веб- чи нативній розробці, але з урахуванням специфіки: використання WebView, взаємодія з плагінами Capacitor, залежність від нативного середовища.

13.2. Типи тестування

1) Модульне (unit testing)

Тестуються окремі компоненти або сервіси:

- логіка додавання завдання;
- обробка даних у сервісі;
- взаємодія з Firestore.

2) Інтеграційне

Перевіряється взаємодія між частинами додатку:

- компонент + сервіс;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 96

- сторінка + маршрутизатор;
- форми + валідація + подія.

3) E2E (end-to-end)

Імітується реальна поведінка користувача:

- відкриття сторінки;
- заповнення форм;
- авторизація та навігація;
- перевірка відображення повідомлень.

4) UI-тестування

Перевіряється правильність відображення елементів інтерфейсу на різних пристроях та орієнтаціях.

5) Перформанс-тестування

Оцінюється швидкодія застосунку, швидкість запуску, рендер сторінок, кількість запитів.

13.3. Інструменти для тестування в Angular/Ionic

Jest або Karma + Jasmine

Для юніт-тестів у середовищі Angular:

- функції, сервіси, pipe-и;
- асинхронні операції з RxJS;
- тестування шаблонів (TestBed).

Cypress

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 97

Підходить для E2E-тестів:

- відкриття сторінок;
- перевірка елементів;
- заповнення форм;
- навігація.

Appium

Інструмент для автоматизованого тестування мобільних застосунків:

- працює з Android та iOS;
- тестує не лише WebView, а й нативні плагіни;
- підтримує WebDriver.

13.4. Приклади юніт-тестів у Angular

```
describe('TaskService', () => {
  let service: TaskService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(TaskService);
  });

  it('should add a task', async () => {
    const task = { id: '1', title: 'Test', isDone: false };

    await service.addTask(task);
    const tasks = await service.getTasks();
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 98

```

        expect(tasks.length).toBe(1);
    });
});

```

13.5. Приклад E2E-тесту з Cypress

```

describe('Login flow', () => {
    it('should log in with correct credentials', () => {
        cy.visit('/login');

        cy.get('input[name=email]').type('test@example.com');
        cy.get('input[name=password]').type('123456');
        cy.get('button').click();
        cy.url().should('include', '/home');
    });
});

```

13.6. Симулятори та реальні пристрої

Android Studio + AVD

- створення емуляторів для різних моделей;
- перевірка UI, жестів, клавіатури, push-повідомлень.

iOS Simulator (через Xcode)

- доступний тільки на macOS;
- дає змогу перевірити інтерфейс iOS, поведінку клавіатури, сповіщення.

Фізичні пристрої

- найнадійніший варіант для перевірки «наживу»;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 99

- особливо важливий для push, біометрії, мережевого стану.

13.7. Firebase Test Lab

- хмарна платформа для автоматичного тестування Android-додатків;
- підтримує Espresso, Robo, Game Loop тести;
- дозволяє тестувати на реальних пристроях Google;
- зручна для інтеграції з CI/CD.

13.8. CI/CD і тестування

Рекомендується автоматизувати процес перевірки:

- запуск тестів перед деплоєм;
- повідомлення про помилки;
- візуальні тести (наприклад, Percy, AppliTools).

CI-платформи:

- GitHub Actions;
- GitLab CI;
- CircleCI.

13.9. Поради з організації тестування

- Тестуйте одразу: не відкладайте на «після MVP»;
- Починайте з найкритичніших функцій;
- Створіть шаблон тестового користувача;
- Не забувайте про випадки без інтернету;
- Регулярно перевіряйте на нових версіях Android/iOS;
- Фіксуйте баги у трекері задач (Jira, GitHub Issues, Trello).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 100

13.10. Висновки

Якісне тестування — це не розкіш, а необхідність. Воно дозволяє:

- підвищити стабільність;
- зменшити кількість скарг від користувачів;
- покращити UX;
- прискорити майбутню розробку;
- зменшити витрати на підтримку.

Для гібридних застосунків, що працюють у WebView, важливо протестувати не лише логіку Angular, а й інтеграцію з нативними функціями, продуктивність і адаптивність інтерфейсу.

Тема 14. Безпека гібридних мобільних додатків

14.1. Вступ

Безпека — критично важлива складова будь-якого мобільного застосунку, особливо якщо він:

- працює з персональними даними користувача;
- виконує автентифікацію;
- використовує платіжні системи;
- має інтеграцію з хмарними сховищами чи API.

Гібридні додатки, як і нативні, можуть стати об'єктом атак, зокрема:

- перехоплення запитів,
- доступ до локального сховища,
- витік токенів доступу,

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 101

- підміна інтерфейсу,
- небезпечні плагіни чи скрипти.

14.2. Основні ризики безпеки

Зберігання токенів у незахищеному вигляді

- Якщо access token зберігається у localStorage або Ionic Storage, його може перехопити злоумисник.

Відсутність шифрування

- Передача персональних даних без HTTPS (наприклад, при POST-запитах).

Відсутність перевірки автентичності

- Необмежений доступ до API, що не перевіряє користувача.

SQL/NoSQL ін'єкції

- Хоч Firestore автоматично захищений, неправильне формування запитів може дати доступ до всіх документів.

Обхід автентифікації через routing

- Якщо маршрути не захищені guard-ами — користувач може зайти в закриті розділи вручну.

14.3. Кращі практики безпеки

HTTPS + CORS

- Усі запити мають відбуватись через **https**;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 102

- Правильно налаштовані **CORS-політики** на бекенді.

Обмеження прав доступу у Firestore

- У Firebase → Firestore → Rules:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /tasks/{taskId} {
      allow read, write: if request.auth != null &&
request.auth.uid == resource.data.userId;
    }
  }
}
```

Захист маршрутів

```
import { canActivate, redirectUnauthorizedTo } from
'@angular/fire/auth-guard';

{ path: 'home', component: HomePage, ...canActivate(()
=> redirectUnauthorizedTo(['login'])) }
```

Біометрія + PIN

Використовуйте плагіни Capacitor:

```
import { NativeBiometric } from
'capacitor-native-biometric';
```

Для доступу до чутливої інформації — авторизація через Touch ID / Face ID.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 103

14.4. Шифрування локального сховища

Небезпечно:

```
this.storage.set('token', token); // зберігається у відкритому вигляді
```

Безпечніше:

- Використовуйте шифрування перед збереженням;
- Плагіни на кшталт @capacitor-community/sqlite із шифруванням;
- capacitor-secure-storage-plugin для Android/iOS.

14.5. Захист токенів та сесій

- Не зберігайте `access_token` надовго;
- Використовуйте **short-lived tokens + refresh token**;
- Зберігайте токен у Memory або захищеному сховищі;
- Автоматичне оновлення сесії через Firebase SDK:

```
onAuthStateChanged(auth, user => {
  // відновлення сесії
});
```

14.6. Перевірка автентичності запитів

Усі запити до API мають:

- містити токен користувача;
- перевіряти, чи дійсний цей токен (на бекенді);

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 104

- повертати 401 при відсутності/некоректності токена.

14.7. Захист від XSS і небезпечних скриптів

Уникайте:

- `innerHTML`;
- `eval()`, `new Function()`.

Angular автоматично **санітує** дані, але ручна вставка HTML має бути під контролем:

```
<div [innerHTML]="value | safeHtml"></div>
```

14.8. Обфускація коду та захист збірки

Для продакшну:

```
ionic build --prod
```

Це:

- мінімізує JS/CSS;
- видаляє дебаг-інформацію;
- запобігає зворотному інжинірингу.

Можна також додатково:

- шифрувати дані конфігурації;
- приховати назви змінних через обфускатори (`javascript-obfuscator`).

14.9. Firebase App Check

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 105

Firebase App Check захищає бекенд-сервіси від доступу з неавторизованих застосунків:

- визначає, чи запит походить із вашого застосунку;
- блокує сторонні скрипти або фальшиві клієнти.

Увімкнення:

- Firebase Console → App Check;
- Генерація App Check Token;
- Підключення SDK у додаток.

14.10. Висновки

Безпека — це не одноразове завдання, а постійний процес. Гібридні мобільні застосунки можуть бути **так само захищеними**, як і нативні, якщо дотримуватись таких принципів:

- не зберігати критичні дані у відкритому доступі;
- правильно обмежувати доступ до бази;
- захищати маршрути, сесії, API;
- шифрувати та оновлювати токени;
- проводити періодичні перевірки на вразливості.

Тема 15. Розгортання гібридних мобільних додатків. CI/CD

15.1. Вступ

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 106

Після завершення розробки мобільного застосунку важливо правильно організувати процес **збірки, публікації та оновлення**. Для цього використовуються:

- нативні інструменти Android/iOS;
- платформи розгортання (Google Play, App Store, TestFlight, Firebase App Distribution);
- сервіси автоматизації CI/CD (GitHub Actions, Bitrise, Codemagic, AppCenter).

Розгортання гібридного додатку потребує адаптації вебчастини до нативної обгортки — через Capacitor або Cordova.

15.2. Підготовка до розгортання

Білд Angular-проекту:

```
ionic build --prod
```

Синхронізація з Capacitor:

```
npm cap sync android
npm cap sync ios
```

Відкриття проекту:

```
npm cap open android
npm cap open ios
```

Після цього зміни в `www` більше не зачіпають платформу автоматично — потрібен `ionic build+cap copy` або `cap sync`.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 107

15.3. Збірка Android

Підписування .apk/.aab:

1. Створення ключа:

```
keytool -genkey -v -keystore release-key.keystore -alias
my-key-alias -keyalg RSA -keysize 2048 -validity 10000
```

2. Додавання до android/app/build.gradle:

```
signingConfigs {
    release {
        storeFile file("release-key.keystore")
        storePassword "your_password"
        keyAlias "my-key-alias"
        keyPassword "your_password"
    }
}
```

Збірка релізної версії:

```
cd android
./gradlew assembleRelease
```

Файл app-release.apk буде готовий до завантаження на Google Play.

15.4. Збірка iOS

1. Відкриття у Xcode:

```
npx cap open ios
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 108

2. Створення профілю:

- обліковий запис Apple Developer;
- provisioning profile + certificate.

3. Архівація → Export → .ipa або завантаження через Xcode Organizer до TestFlight / App Store Connect.

15.5. CI/CD: що це і навіщо?

CI/CD (Continuous Integration / Continuous Delivery) — це автоматизація:

- тестування;
- збірки;
- публікації застосунку.

Переваги:

- економія часу;
- зниження кількості помилок;
- стабільний реліз-процес.

15.6. Платформи CI/CD для мобільних застосунків

Платформа	Особливості
GitHub Actions	Безкоштовна, інтегрується з репозиторієм
Bitrise	Окрема мобільна платформа, з GUI
Codemagic	Нативна підтримка Flutter/Ionic
App Center	Від Microsoft, з аналітикою
Firebase App Distribution	Швидкий внутрішній реліз

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 109

15.7. Приклад GitHub Actions для Ionic + Android

```
name: Build Android
```

```
on:
```

```
  push:
```

```
    branches:
```

```
      - main
```

```
jobs:
```

```
  build:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - uses: actions/checkout@v2
```

```
      - name: Install dependencies
```

```
        run: npm ci
```

```
      - name: Build Ionic app
```

```
        run: npm run build
```

```
      - name: Sync with Capacitor
```

```
        run: npx cap sync android
```

```
      - name: Build APK
```

```
        run: |
```

```
          cd android
```

```
          ./gradlew assembleRelease
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 110

15.8. Публікація на Google Play

1. Зареєструйтесь у Google Play Console (25\$ одноразово);
2. Створіть новий застосунок;
3. Завантажте .aab файл;
4. Додайте опис, скріншоти, політику конфіденційності;
5. Підтвердьте реліз.

15.9. Публікація на Firebase App Distribution

1. Додайте Firebase до проєкту;
2. Встановіть CLI:

```
npm install -g firebase-tools
firebase login
```

3. Завантаження збірки:

```
firebase appdistribution:distribute path/to/app.apk \
  --app your-app-id \
  --groups "testers"
```

15.10. Висновки

Розгортання мобільного додатку — це не лише технічна дія, а фінальний етап життєвого циклу проєкту. Щоб зробити його стабільним і повторюваним:

- автоматизуйте збірку через CI/CD;
- використовуйте профілі підпису;
- тестуйте на реальних пристроях;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 111

- використовуйте Firebase Distribution або TestFlight для внутрішнього релізу;
- готуйте всі матеріали (опис, скріншоти, політику) заздалегідь.

ЛАБОРАТОРНА РОБОТА №6

Тема: Тестування, безпека та розгортання гібридного мобільного додатку

Мета роботи: Навчитись: створювати юніт-тести та E2E-тести; захищати дані користувача, обмежувати доступ до Firestore; підготувати додаток до продакшн-збірки та ознайомитись із CI/CD.

Короткі теоретичні відомості:

- **Unit testing** — перевірка окремих функцій, сервісів, ріп-ів.
- **E2E testing** — імітація дій користувача у браузері.
- **CI/CD** — автоматизація збірки, тестування та публікації застосунку.
- **Firestore Security Rules** — правила доступу до бази.
- **Production build** — оптимізована збірка з мінімізацією і шифруванням.

Хід виконання лабораторної роботи:

КРОК 1. Додайте юніт-тест для TaskService

1. Створіть у папці src/app/services файл task.service.spec.ts.
2. Напишіть тест для функції addTask():

```
describe('TaskService', () => {
  let service: TaskService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(TaskService);
  });
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 112

```

it('should add a task to the list', async () => {
  const task = { id: '1', title: 'Test Task', isDone: false };
  await service.addTask(task);
  const tasks = await service.getTasks();
  expect(tasks).toContain(jasmine.objectContaining(task));
});
});

```

КРОК 2. Створіть E2E-тест за допомогою Cypress

1. Встановіть Cypress:
npm install cypress --save-dev
2. Додайте тест до cypress/e2e/login.cy.ts:

```

describe('Login flow', () => {

  it('should login and redirect to home', () => {

    cy.visit('/login');

    cy.get('input[name=email]').type('test@example.com');

    cy.get('input[name=password]').type('123456');

    cy.get('ion-button').click();

    cy.url().should('include', '/home');

  });

});

```

КРОК 3. Налаштуйте Firestore Security Rules

У Firebase Console → Firestore → Rules:

```

rules_version = '2';

service cloud.firestore {

  match /databases/{database}/documents {

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 113

```

match /tasks/{taskId} {

  allow read, write: if request.auth != null && request.auth.uid ==
resource.data.userId;

}

}

}

```

КРОК 4. Увімкніть HTTPS і App Check

1. У Firebase → App Check → Увімкніть перевірку.
2. Для Web додайте App Check у main.ts:

```

import { initializeAppCheck, ReCaptchaV3Provider } from 'firebase/app-check';

initializeAppCheck(app, {

  provider: new ReCaptchaV3Provider('your-site-key'),

  isTokenAutoRefreshEnabled: true

});

```

КРОК 5. Підготовка до продакшн-збірки

```
ionic build --prod
```

```
npm cap sync
```

- Перевірте, що не зберігаються токени у localStorage.
- Всі секрети винесені у environment.prod.ts.

КРОК 6. Створіть CI-скрипт (GitHub Actions)

```
name: Build & Test
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 114

on: [push]

jobs:

test:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2
- run: npm ci
- run: npm run test -- --watch=false --browsers=ChromeHeadless
- run: npm run build -- --prod

Вимоги до оформлення звіту:

- Скриншоти:
 - виконаного юніт-тесту
 - виконаного E2E-тесту
 - продакшн-збірки (www/, android/, ios/)
 - Firebase Rules
 - CI-звіту (GitHub Actions або інший)
- Код task.service.spec.ts
- Код E2E-тесту
- Посилання на репозиторій (якщо є CI)
- Висновки

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 115

Контрольні запитання:

1. У чому різниця між unit та E2E тестами?
2. Які є способи захисту доступу до Firestore?
3. Що таке App Check і навіщо він?
4. Як відрізняється dev-збірка від prod-збірки в Angular?
5. Що таке CI/CD і які його переваги?

СПИСОК ЛІТЕРАТУРИ

1. Mozilla Developer Network. JavaScript documentation [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
2. Angular. Angular official documentation [Електронний ресурс]. – Режим доступу: <https://angular.io/docs>
3. Ionic Framework. Documentation [Електронний ресурс]. – Режим доступу: <https://ionicframework.com/docs>
4. Capacitor Documentation – Building cross-platform native apps with web technology [Електронний ресурс]. – Режим доступу: <https://capacitorjs.com/docs>
5. Firebase. Documentation [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs>
6. Кравченко І.І. Розробка гібридних мобільних додатків: навчальний посібник. – Харків: ХНУРЕ, 2020. – 114 с.
7. Гребенюк П.О. Основи роботи з Angular: навч. посіб. – Київ: КНУ ім. Тараса Шевченка, 2021. – 96 с.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 116

8. Сидоренко А.В. Безпека мобільних додатків. – Львів: ЛНУ імені Івана Франка, 2019. – 88 с.
9. Мельник С.А. Основи веб-програмування: навч. посіб. – Київ: НАУ, 2022. – 156 с.
10. Шевченко І.В. Тестування програмного забезпечення: навч. посіб. – Харків: УПА, 2020. – 144 с.
11. Резнік Ю.М. Основи проєктування мобільних додатків. – Одеса: ОНАХТ, 2021. – 106 с.
12. Freeman A., Sanderson A. Pro Angular 9. – Apress, 2020. – 760 p.
13. Hart D. Ionic 6 Projects: Build exciting cross-platform mobile apps with Ionic, Angular, and Capacitor. – Packt Publishing, 2022. – 420 p.
14. Majors T. Firebase Essentials: Learn how to use Firebase to quickly build real-time applications. – O'Reilly Media, 2021. – 256 p.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 117

ДОДАТКИ

ТЕСТОВІ ПИТАННЯ

№ п/п	Текст завдання	Варіанти відповідей
1	Що таке гібридний мобільний застосунок?	А. Застосунок, написаний на C++ В. Застосунок, що працює лише онлайн С. Застосунок, що використовує вебтехнології, але запускається нативно D. Ігровий застосунок
2	Яка основна перевага гібридного підходу?	А. Низька продуктивність В. Можливість створити одну кодову базу для кількох платформ С. Неможливість доступу до камери D. Відсутність підтримки магазинів застосунків
3	Яка роль Capacitor у мобільній розробці?	А. Запускає сервер для API В. Надає нативні плагіни для вебзастосунків С. Створює дизайни UI D. Тестує застосунки
4	Що таке плагін у Capacitor?	А. Тема оформлення В. Код для монетизації С. Розширення, що дозволяє взаємодіяти з нативним кодом D. Аналізатор продуктивності
5	Що забезпечує Firebase Authentication?	А. Хмарне сховище В. Інструменти UI С. Механізми автентифікації користувачів D. Обробку аудіо
6	Який із способів автентифікації підтримує Firebase?	А. QR-код лише В. Google, Email/Password, Анонімна автентифікація С. Bluetooth D. Тільки SMS
7	Яку мову програмування зазвичай використовують для гібридних застосунків з Ionic?	А. Java В. Swift С. JavaScript/TypeScript D. Python

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 118

8	Яка технологія лежить в основі інтерфейсу гібридного застосунку?	A. OpenGL B. HTML/CSS C. C# D. PHP
9	Яка команда встановлює Capacitor у проєкт?	A. `npm install @capacitor/core @capacitor/cli` B. `ionic install capacitor` C. `npx add capacitor` D. `install capacitor --cli`
10	Що потрібно зробити після ініціалізації Capacitor, щоб відобразити проєкт у Xcode?	A. Відкрити браузер B. Запустити Android Studio C. Виконати `npx cap open ios` D. Змінити тему застосунку
11	Яка команда створює нову сторінку в Ionic з Angular?	A. `ionic generate component` B. `ionic generate page` C. `ionic new page` D. `ionic page create`
12	Що таке `ion-router-outlet`?	A. Панель навігації B. Тег для стилів C. Компонент для відображення сторінок залежно від маршруту D. Аналог ViewContainer
13	Що з нижченаведеного є найкращою UX-практикою?	A. Перевантажений інтерфейс B. Однакова кнопка для всіх дій C. Чіткі візуальні ієрархії D. Всі кнопки одного кольору
14	Що таке адаптивний дизайн у мобільному застосунку?	A. Дизайн лише під Android B. Фіксований розмір елементів C. Підлаштування під розмір екрана D. Ігнорування розміру екрану
15	Що означає аббревіатура i18n?	A. Internet to native B. Internationalization C. Integration layer 18 D. Interactive node
16	Яка директива Angular допомагає з локалізацією?	A. `ngSwitch` B. `ngIf` C. `i18n` D. `ngStyle`
17	Який плагін дозволяє використовувати камеру у Capacitor?	A. `@capacitor/filesystem` B. `@capacitor/camera`

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 119

		C. `@ionic/camera` D. `native-camera`
18	Що таке geolocation у мобільних застосунках?	A. Обчислення висоти тексту B. Зчитування кольору пікселя C. Визначення координат пристрою D. Відправка SMS
19	Коли викликається `ionViewDidEnter`?	A. Перед ініціалізацією компонента B. Після кожного входу на сторінку C. Після натискання кнопки D. Після реєстрації
20	Який життєвий цикл Angular викликається після побудови компонента?	A. `ngOnDestroy` B. `ngIf` C. `ngAfterViewInit` D. `ngOnInit`
21	Що потрібно для логіну з Google у Firebase?	A. Тільки Email B. Додати Google як провайдера автентифікації C. Встановити AngularFireAnalytics D. Запустити локальний сервер
22	Який метод авторизації підтримується у Firebase?	A. Тільки телефон B. Email/Password, Google, Anonymous C. Тільки GitHub D. Тільки OAuth 1.0
23	Що дозволяє відстежувати Firebase Analytics?	A. Кількість лайків B. Повідомлення в чаті C. Події та поведінку користувачів D. Кількість натискань на іконки
24	Як додати подію в Firebase Analytics у застосунку?	A. `analytics.logEvent('event_name')` B. `firebase.log()` C. `track.event()` D. `record('event')`
25	Яка технологія використовується Firebase для push-сповіщень?	A. WebRTC B. FCM C. XML-RPC D. Firebase Shell
26	Який метод дозволяє надіслати сповіщення?	A. `sendAlert()` B. `pushNow()` C. `sendNotification()` D. `sendMessage()`
27	Що таке MapLibre?	A. Компонент Firebase B. Нативна бібліотека Android

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 120

		С. Бібліотека для інтеграції карт з відкритим кодом D. Протокол шифрування
28	Яка структура даних використовується для збереження маршрутів у MapLibre?	A. CSV B. XML C. GeoJSON D. SQL
29	Що таке Service в Angular/Ionic?	A. UI-компонент B. Файл стилів C. Логічний клас, який використовується для спільних функцій D. Конфігурація проекту
30	Що таке Dependency Injection у контексті Angular?	A. Запуск програми B. Вивід UI C. Механізм передавання залежностей у компоненти D. Інтерфейс кнопки
31	Який з наступних тегів використовується для створення кнопки в Ionic?	A. ion-text B. ion-card C. ion-button D. ion-tab
32	Що таке lifecycle hook у Angular?	A. Зовнішній API B. Функція, яка викликається на певному етапі життя компонента C. CSS-клас D. Тестовий модуль
33	Який плагін Capacitor дозволяє зчитувати файли?	A. @capacitor/splash-screen B. @capacitor/filesystem C. @capacitor/haptics D. @capacitor/toast
34	Що таке T9n?	A. Автоматичний переклад B. Скорочення від 'Translation' C. Скорочення від 'Transformation' D. Тип локалізації
35	Як можна додати Firebase до Angular-проекту?	A. firebase init B. npm install firebase C. npx firebase connect D. firebase add angular
36	Яка з наведених платформ дозволяє	A. Unity B. Xamarin C. React Native

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 121

	створювати гібридні мобільні застосунки?	D. Ionic"
37	Що таке `Capacitor Bridge`?	A. Міст для Bluetooth-з'єднання B. Інструмент для переходу між сторінками C. Механізм виклику нативного коду з JavaScript D. Бібліотека стилів
38	Яка перевага використання Firebase Authentication?	A. Підтримка лише локальної авторизації B. Потрібен власний сервер C. Швидка інтеграція кількох способів авторизації D. Використання лише паролів
39	Що робить `ion-loading` у Ionic?	A. Відображає карту B. Показує індикатор завантаження C. Оновлює компонент D. Запускає push-сповіщення
40	Що означає `lazy loading` у маршрутизації Angular?	A. Швидке завантаження ресурсів B. Видалення непотрібних стилів C. Завантаження модулів лише за потреби D. Підключення шрифтів
41	Який з наступних методів використовується для отримання позиції користувача у Capacitor?	A. `getLocation()` B. `getCurrentPosition()` C. `getUserCoords()` D. `fetchPosition()`
42	Що таке `Splash Screen` у мобільному застосунку?	A. Резервна копія B. Заставка при запуску застосунку C. Помилка з'єднання D. Меню навігації
43	Що з наступного є обов'язковим для налаштування Push-сповіщень через FCM?	A. Лише токен користувача B. Інтернет-з'єднання C. Firebase проект і ключ сервера D. Доступ до камери
44	Яке сховище рекомендується для зберігання великих об'ємів структурованих даних офлайн?	A. LocalStorage B. SessionStorage C. IndexedDB D. Cookies
45	Що таке `ion-segment` у Ionic?	A. Форма логіну B. Компонент для вибору між категоріями

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 122

		С. Віджет годинника D. Контейнер з фоном
46	Що таке Firebase Cloud Functions?	A. Бібліотека стилів B. Функції, які запускаються у браузері C. Функції, що запускаються у хмарі у відповідь на події D. Компонент Angular
47	Що таке '@Injectable()' в Angular?	A. Маркер для сервісів, які можуть впроваджуватись B. Маркер компонента C. Фільтр D. Компонент форми
48	Що з цього забезпечує навігацію між сторінками в Ionic?	A. 'RouterLink' B. 'RouterPage' C. 'NavigateTo' D. 'RouteSwitch'
49	Який lifecycle hook викликається один раз при ініціалізації компонента?	A. 'ngAfterViewInit' B. 'ngOnDestroy' C. 'ngOnInit' D. 'ionViewDidEnter'
50	Для чого використовується Firebase Analytics у мобільному застосунку?	A. Оновлення інтерфейсу B. Відправка повідомлень C. Відстеження дій користувачів D. Блокування ботів
51	Що таке 'ion-modal'?	A. Сторінка реєстрації B. Компонент для спливаючих вікон C. Push-компонент D. Анімація
52	Який тип автентифікації забезпечує найменший поріг входу для користувача?	A. Email/Password B. Анонімна автентифікація C. OAuth D. SAML
53	Що з цього дозволяє кешувати карти для offline-режиму?	A. Firebase B. MapLibre з попереднім завантаженням тайлів C. Capacitor CLI D. Google Forms
54	Яке з полів Angular форми є обов'язковим для валідації?	A. 'required' B. 'submit' C. 'value' D. 'check'

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 123

55	Що таке `ion-tabs`?	А. Компонент для створення вкладок В. Список елементів С. Таблиця даних D. Поле пошуку
----	---------------------	---

ОСНОВНІ КОМАНДИ JS, ANGULAR ТА IONIC FRAMEWORK

Ініціалізація

- `npm init` – ініціалізувати новий проєкт Node.js
- `npm install <package>` – встановити пакет
- `npm uninstall <package>` – видалити пакет
- `ng new` – створити новий проєкт Angular
- `ng add @ionic/angular` – додати підтримку Ionic до Angular-проєкту
- `ng serve` – запустити Angular-проєкт у режимі розробки
- `ionic start` – створити новий проєкт Ionic
- `ionic serve` – запустити застосунок у браузері
- `ionic build` – зібрати застосунок для продакшну
- `npx cap add android` – додати Android-платформу
- `npx cap open android` – відкрити Android-проєкт у Android Studio
- `npx cap sync` – синхронізувати платформи з веб-кодом

Дані

- `const arr = [1, 2, 3]` – створити масив
- `Array.from()` – створити масив із іншого об'єкта
- `arr.map()` – застосувати функцію до кожного елемента масиву
- `arr.filter()` – відфільтрувати масив
- `arr.reduce()` – звести масив до одного значення
- `arr.includes(value)` – перевірити наявність значення в масиві
- `Math.random()` – згенерувати випадкове число
- `Math.floor()` – округлити до меншого цілого
- `new Date()` – створити об'єкт дати

Статистика та обчислення

- `Math.max(...arr)` – найбільше значення в масиві

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 124

- `Math.min(...arr)` – найменше значення
- `arr.reduce((a, b) => a + b) / arr.length` – середнє значення
- `Math.sqrt(n)` – квадратний корінь
- `Math.pow(a, b)` – піднесення до степеня

Angular: створення і логіка

- `ng generate component component-name` – створити компонент
- `ng generate service service-name` – створити сервіс
- `@Input()` – отримати дані від батьківського компонента
- `@Output()` – передати подію з дочірнього компонента
- `ngOnInit()` – хук ініціалізації компонента
- `[(ngModel)]` – двостороннє зв'язування даних
- `*ngIf` – умовне відображення елемента
- `*ngFor` – цикл по елементах

Angular: маршрутизація

- `RouterModule.forRoot(routes)` – налаштування маршрутизатора
- `[routerLink]='"/home"'` – прив'язка переходу у шаблоні
- `this.router.navigate(['/home'])` – програмний перехід
- `canActivate` – захист маршруту (автентифікація)

Ionic: компоненти

- `<ion-button>` – кнопка
- `<ion-input>` – поле вводу
- `<ion-list>` – список
- `<ion-item>` – елемент списку
- `<ion-alert>` – модальне повідомлення
- `<ion-modal>` – модальне вікно
- `<ion-loading>` – індикатор завантаження
- `<ion-tabs>` – таби (нижня навігація)
- `<ion-menu>` – бокове меню

Нативні функції (Capacitor)

- `Camera.getPhoto()` – відкрити камеру / вибрати фото
- `Geolocation.getCurrentPosition()` – отримати координати користувача

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/1 22.00.Б/ВК-2025
	Екземпляр № 1	Арк 124 / 125

- `PushNotifications.register()` – зареєструвати пристрій для сповіщень
- `Haptics.impact()` – вібрація (зворотний зв'язок)
- `Network.getStatus()` – перевірка підключення до інтернету

Тестування

- `ng test` – запуснути юніт-тести
- `ng e2e` – запуснути end-to-end тести
- `describe()`, `it()`, `expect()` – структура юніт-тестів
- `cy.visit()`, `cy.get()` – основні команди Cypress для E2E
- `TestBed.configureTestingModule()` – налаштування тестового модуля
- `fixture.detectChanges()` – ініціалізація компонента в тесті

CI/CD

- `npm run build --prod` – продакшн-збірка
- `firebase deploy` – розгортання на Firebase
- `ionic deploy` – оновлення застосунку (якщо використовується Appflow)
- `.github/workflows/*.yaml` – конфігурація автоматичного CI

Довідка і підтримка

- `console.log()` – вивід значень у консоль
- `console.error()` – вивід помилки
- `ng help` – список команд Angular CLI
- `ionic --help` – довідка по командам Ionic