

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Декан факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО
« 22 » 2025 р.

ЗАТВЕРДЖУЮ
Проректор
з науково-педагогічної роботи
Віктор ОСТРОВЕРХОВ
« 22 » 2025 р.

ЗАТВЕРДЖУЮ
Директор навчально-
наукового інституту цювітніх
освітніх технологій
Святослав ПИТЕЛЬ
« 22 » 2025 р.



РОБОЧА ПРОГРАМА

з дисципліни

«Інженерія кросплатформного програмного забезпечення»

ступінь вищої освіти: магістр
галузь знань – F «Інформаційні технології»
спеціальність – F2 «Інженерія програмного забезпечення»
освітньо-професійна програма – «Інженерія програмного забезпечення»

Кафедра комп'ютерних наук

Форма навчання	Курс	Семестр	Лекції (год.)	Практ. (год.)	ІРС (год.)	Тренінг (год.)	СРС (год.)	Разом (год.)	Екзамен (сем.)
Денна	1	2	30	14	4	6	96	150	2
Заочна	1	2	8	4			138	150	2

Тернопіль – ЗУНУ
2025

Робоча програма розроблена доцентом кафедри комп'ютерних наук,
к.т.н., Іриною СТАСІВ.

Робочу програму затверджено на засіданні кафедри комп'ютерних
наук, протокол № 1 від 26 серпня 2025 р

Завідувач кафедри
д.т.н., професор



Андрій ПУКАС

Гарант ОП
д.т.н., професор



Андрій ПУКАС

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Інженерія кросплатформного програмного забезпечення»

1. Опис дисципліни «Інженерія кросплатформного програмного забезпечення»

Дисципліна «Інженерія кросплатформного програмного забезпечення»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 5	Галузь знань – F «Інформаційні технології»	Статус дисципліни: вибіркова Мова навчання: Українська
Кількість залікових модулів – 4	Спеціальність – F2 «Інженерія програмного забезпечення»	Рік підготовки: Денна – 1. Заочна – 1. Семестр: Денна – 2. Заочна – 2 Лекції: Денна – 30 год. Заочна – 8 год.
Кількість змістових модулів – 2	Ступінь вищої освіти: магістр	Практичні роботи: Денна – 14 год. Заочна – 4 год. Самостійна робота: Денна – 96 годин, Заочна – 138 год.
Загальна кількість годин – 150		Тренінг: Денна – 6 годин Індивідуальна робота: Денна – 4 години
Тижневих годин – 10 год., з них аудиторних – 3 год.		Вид підсумкового контролю – екзамен

2. Мета й завдання вивчення дисципліни «Інженерія кросплатформного програмного забезпечення»

2.1. Мета вивчення дисципліни

Метою курсу «Інженерія кросплатформного програмного забезпечення» є набуття навиків проектування і реалізації ПЗ, що здатне функціонувати на більше як одній платформі.

Вивчення курсу «Інженерія кросплатформного програмного забезпечення» вимагає цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях та практичних заняттях, самостійної роботи та виконання індивідуальних завдань.

2.2. Завдання вивчення дисципліни

Завданням вивчення дисципліни «Інженерія кросплатформного програмного забезпечення» є набуття навиків проектування і реалізації ПЗ, що здатне функціонувати на більше як одній платформі.

Вивчення дисципліни «Інженерія кросплатформного програмного забезпечення» забезпечує у здобувачів здатність до абстрактного мислення, аналізу та синтезу; удосконалювати свої навички на основі аналізу попереднього досвіду; генерувати нові ідеї (креативність); аналізувати предметні області, формувати, аналізувати та моделювати вимоги до програмного забезпечення; розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення; розробляти і координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення.

2.3. Результати навчання

У результаті навчання студент повинен:

- розуміти поняття апаратної і програмної платформ;
- знати принципи організації коду в такий спосіб, щоби щонайбільша кількість модулів могла бути використаною без змін при збиранні на інших апаратних і програмних платформах;
- вміти користуватися кросплатформними засобами збирання виконуваних файлів;
- розуміти поняття віртуальної машини та технології розробки програмного забезпечення для віртуальної машини;
- розуміти властивості середовища виконання браузерів і хмарних баз даних та способи доступу до них;
- отримати практичний досвід створення власного кросплатформного додатку.

3. Програма навчальної дисципліни «Інженерія кросплатформного програмного забезпечення»

Змістовий модуль 1. Технології розробки кросплатформних додатків

Тема 1. Особливості кросплатформної розробки програм

Тема 2. Основні поняття .NET MAUI

Тема 3. Створення мобільних та настільних програм за допомогою .NET MAUI

Тема 4. Основи роботи з Flutter

Тема 5. Розробка кросплатформних додатків на Flutter

Тема 6. Основні компоненти та особливості фреймворка React Native

Тема 7. Кросплатформний фреймворк з відкритим кодом Ionic

Тема 8. Розробка додатків на Ionic Angular

Тема 9. Технологія створення багатоплатформних проєктів Kotlin Multiplatform.

Тема 10. Створення багатоплатформних проєктів Kotlin

Змістовий модуль 2.

Спеціальні можливості кросплатформної розробки

Тема 11. Інтеграція кросплатформних додатків із хмарними сервісами

Тема 12. Забезпечення якості кросплатформних застосунків

4. Структура залікового кредиту з дисципліни

«Інженерія кросплатформного програмного забезпечення»

<i>Денна форма навчання</i>	<i>Кількість годин</i>				Контрольні заходи
	Лекції	Практичні заняття	Самостійна робота	Тренінг	
Змістовий модуль 1. Технології розробки кросплатформних додатків					
Тема 1. Особливості кросплатформної розробки програм	2	2	8	2	Усне опитування та тестування
Тема 2. Основні поняття .NET MAUI	4		8		Усне опитування та тестування
Тема 3. Створення мобільних та настільних програм за допомогою .NET MAUI	2	2	8		Усне опитування та тестування
Тема 4. Основи роботи з Flutter	4		8		Усне опитування та тестування
Тема 5. Розробка кросплатформних додатків на Flutter	2	2	8		Усне опитування та тестування
Тема 6. Основні компоненти та особливості фреймворка React Native	2		8		Усне опитування та тестування
Тема 7. Кросплатформний фреймворк з відкритим кодом Ionic	4		8		Усне опитування та тестування
Тема 8. Розробка додатків на Ionic Angular	2	2	8		Усне опитування та тестування
Тема 9. Технологія створення багатоплатформних проєктів Kotlin Multiplatform.	2		8		Усне опитування та тестування
Тема 10. Створення багатоплатформних проєктів Kotlin	2	2	8		Усне опитування та тестування
Змістовий модуль 2. Спеціальні можливості кросплатформної розробки					
Тема 11. Інтеграція кросплатформних	2	2	8	2	Усне опитування та тестування

додатків із хмарними сервісами					
Тема 12. Забезпечення якості кросплатформних застосунків	2	2	8	2	Усне опитування та тестування
Разом	30	14	96	6	

<i>Заочна форма навчання</i>	<i>Кількість годин</i>		
	Лекції	Практичні заняття	Самостійна робота
Змістовий модуль 1. Технології розробки кросплатформних додатків			
Тема 1. Особливості кросплатформної розробки програм	2		10
Тема 2. Основні поняття .NET MAUI	2		12
Тема 3. Створення мобільних та настільних програм за допомогою .NET MAUI		2	12
Тема 4. Основи роботи з Flutter	2		10
Тема 5. Розробка кросплатформних додатків на Flutter		2	12
Тема 6. Основні компоненти та особливості фреймворка React Native			12
Тема 7. Кросплатформний фреймворк з відкритим кодом Ionic			12
Тема 8. Розробка додатків на Ionic Angular			12
Тема 9. Технологія створення багатоплатформних проєктів Kotlin Multiplatform.			12
Тема 10. Створення багатоплатформних проєктів Kotlin			12
Змістовий модуль 2. Спеціальні можливості кросплатформної розробки			
Тема 11. Інтеграція кросплатформних додатків із хмарними сервісами	2		10
Тема 12. Забезпечення якості кросплатформних застосунків			12
Разом	8	4	138

5. Тематика практичних занять

Практичне заняття №1

Тема. Створення найпростішого додатку за допомогою .NET MAUI

Мета. Навчитися створювати додатки за допомогою .NET MAUI

Питання для обговорення

1. Які основні інструменти та фреймворки використовуються для розробки кросплатформних додатків на C#?
2. Які переваги використання .NET MAUI для кросплатформної розробки порівняно з Xamarin?
3. Як працює система компіляції коду для різних платформ у .NET MAUI?

Практичне заняття №2

Тема. Створення додатку на Flutter

Мета. Навчитися використовувати Flutter для кросплатформних додатків

Питання для обговорення

1. Які переваги Flutter порівняно з іншими фреймворками для кросплатформної розробки (React Native, .NET MAUI, Ionic)?
2. Як побудована архітектура Flutter-додатку: роль віджетів, стану (state) та дерева компонентів?
3. Які труднощі можуть виникнути при створенні інтерфейсу, спільного для Android та iOS, і як їх вирішує Flutter?

Практичне заняття №3

Тема. Розробка додатків на Ionic Angular

Мета. Навчитися розробляти найпростіші додатки на Ionic Angular

Питання для обговорення

1. У чому переваги використання Ionic Angular порівняно з чистим Angular для створення мобільних застосунків?
2. Як працює структура проєкту Ionic (папки src, pages, components, services) і яку роль відіграють модулі Angular у цьому середовищі?
3. Як реалізується навігація між сторінками та передача даних у додатку Ionic Angular?

Практичне заняття №4

Тема. Робота з Kotlin Multiplatform

Мета. Здобути навички у роботі з Kotlin Multiplatform

Питання для обговорення

1. Яку головну ідею реалізує підхід Kotlin Multiplatform і як він відрізняється від традиційної кросплатформної розробки (наприклад, Flutter чи React Native)?
2. Які типи коду можна спільно використовувати між платформами в Kotlin Multiplatform, а які — потрібно реалізовувати окремо?

3. Які інструменти та плагіни використовуються для створення та налаштування Kotlin Multiplatform-проектів у середовищі IntelliJ IDEA або Android Studio?

6. Самостійна робота

Самостійна робота є основним засобом засвоєння студентами навчального матеріалу в позааудиторний час без участі викладача. Обсяг і зміст самостійної роботи визначається робочою програмою та робочим планом в межах встановленого обсягу годин із навчальної дисципліни, методичними вказівками викладача.

Самостійна робота студента забезпечується системою навчально-методичних засобів: конспектами лекцій викладача, підручниками, навчальними та методичними посібниками, монографічною літературою і періодикою, а також засобами самоконтролю (тестами, типовим пакетом контрольних завдань).

Навчальний матеріал, передбачений навчальним планом для засвоєння студентом в процесі самостійної роботи, вноситься на підсумковий контроль поряд з навчальними матеріалом, який опрацьовувався при проведенні навчальних занять.

Викладач проводить діагностику якості самостійної роботи студента на індивідуальних заняттях. Вони проводяться в позааудиторний час за окремим графіком, складеним кафедрою.

Перелік тем для самостійної роботи

1. Інструменти керування станом у Flutter (Provider, Riverpod, Bloc)
2. Використання баз даних у кросплатформних додатках (SQLite, Hive, Realm)
3. Інтеграція REST API у Flutter та React Native
4. Робота з JSON та об'єктною серіалізацією в .NET MAUI і Kotlin Multiplatform
5. Використання нативних плагінів і API у Flutter та React Native
6. Механізми навігації в різних фреймворках (Flutter Navigator, Ionic Router, MAUI Shell)
7. Управління станом додатка та життєвим циклом компонентів
8. Використання Dependency Injection у кросплатформних проєктах
9. Основи роботи з GraphQL у мобільних застосунках
10. Локалізація та інтернаціоналізація кросплатформних додатків
11. Організація зберігання даних у хмарі (Firebase Firestore, AWS Amplify, Supabase)
12. Реалізація push-сповіщень у Flutter, Ionic та React Native
13. Безпека даних у кросплатформних додатках (шифрування, токени, авторизація OAuth2)
14. Тестування UI та автоматизація перевірок у Flutter і .NET MAUI
15. Створення адаптивного інтерфейсу під різні розміри екранів і платформи
16. Робота з камерами, сенсорами та геолокацією у мобільних додатках

17. Використання Git і GitHub у спільній розробці кросплатформних проєктів
18. Публікація додатків у Google Play та App Store: основні вимоги й процеси
19. Використання Docker і контейнеризації у процесі розробки кросплатформних сервісів
20. Підходи до створення Progressive Web Apps (PWA) на базі Angular або React

5. Тренінг з дисципліни

Тренінг (training) – це запланований процес модифікації (зміни) відношення, знання чи поведінкових навичок того, хто навчається, через набуття навчального досвіду з тим, щоб досягти ефективного виконання поставлених завдань в одному із видів діяльності.

Порядок проведення тренінгу:

Тема: «Розробка кросплатформного застосунку від ідеї до готового продукту»

Мета: закріпити знання про фреймворки, навчитися працювати в команді, розробляти багатоплатформові додатки та інтегрувати зовнішні сервіси.

Структура тренінгу:

Частина 1. Постановка завдання та вибір технології (ознайомлення з метою проєкту (наприклад, мобільний додаток для управління завданнями чи міні-магазин); формування груп по 3–5 студентів; аналіз вимог: які платформи підтримуються, потрібна інтеграція з API, база даних, push-сповіщення). Завдання: обрати фреймворк для реалізації проєкту (.NET MAUI, Flutter, Ionic, Kotlin Multiplatform).

Частина 2. Планування та дизайн архітектури (створення плану проєкту: модулі, структура коду, UI/UX дизайн; вибір архітектурного підходу (MVVM, Redux, Clean Architecture). Завдання: розробити макет інтерфейсу та схему взаємодії між модулями.

Частина 3. Реалізація основної функціональності (створення базових екранів, реалізація основної логіки; підключення бази даних і зовнішніх сервісів (Firebase, REST API). Завдання: реалізувати ключові функції додатку (логіні, додавання/редагування даних, відображення списків).

Частина 4. Інтеграція та розширені можливості (додавання інтерактивних елементів, push-сповіщень, роботи з мультимедіа, геолокації). Завдання: реалізувати 1–2 розширені функції (наприклад, карта, камера, повідомлення).

Частина 5. Тестування та налагодження (тестування UI та логіки на різних платформах; профілювання продуктивності, виправлення помилок). Завдання: підготувати проєкт до демонстрації та перевірки на всіх платформах.

Частина 6. Презентація та підсумки (демонстрація готових проєктів групами; обговорення сильних і слабких сторін рішень, труднощів під час розробки; підведення підсумків: що було вивчено, які навички набуті)

Орієнтована тематика завдань на тренінг: створення повноцінного кросплатформного застосунку для Android, iOS і Windows (наприклад, додаток для: управління завданнями/планування часу, особистого бюджету, простого e-commerce магазину, соціальної платформи для обміну повідомленнями).

6. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Інженерія кросплатформного програмного забезпечення» використовуються наступні методи оцінювання навчальної роботи студентів:

- поточне опитування;
- модульне тестування та опитування;
- оцінювання результатів виконання самостійної роботи;
- оцінка за виконані завдання під час тренінгу;
- екзамен;
- інші види індивідуальних та групових завдань.

8. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Інженерія кросплатформного програмного забезпечення» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2	Модуль 3	Модуль 4
20%	20 %	5 %	15%	40%
Поточне оцінювання	Модульний контроль	Тренінги	Самостійна робота	Екзамен
Оцінка за поточне оцінювання визначається, як середнє арифметичне з оцінок, отриманих під час усного опитування на заняттях	Середнє арифметичне із оцінок, отриманих за виконання практичних робіт 1-4	Визначається як єдина оцінка, отримана під час презентації проєкту, розробленого під час тренінгу	Визначається як єдина оцінка по 100-бальній шкалі за результат самостійної роботи за обраною тематикою	3 теоретичні питання для усної відповіді (максимально 100 балів)

Критерії оцінювання відповіді здобувача під час усного опитування

А (відмінно) – студент вільно володіє матеріалом усіх тем курсу; демонструє глибоке розуміння принципів кросплатформної розробки; вміє порівнювати фреймворки (.NET MAUI, Flutter, React Native, Ionic, Kotlin Multiplatform); логічно, послідовно й самостійно формулює відповіді; наводить приклади з практики або власного досвіду.

В (дуже добре) – знання повні та правильні; студент упевнено орієнтується в темах курсу, але окремі пояснення не мають достатньої глибини або деталізації; приклади наведено, але частково шаблонно.

С (добре) – студент володіє основними поняттями курсу; правильно пояснює більшість питань, але відповіді неповні або нечіткі; є труднощі з порівнянням технологій чи обґрунтуванням вибору фреймворку.

Д (задовільно) – студент має загальне уявлення про основні теми, але знання поверхові; дає короткі або частково правильні відповіді; плутає терміни або не завжди розуміє практичне застосування технологій.

Е (достатньо для зарахування) – демонструє мінімальне розуміння курсу; відповіді неповні, фрагментарні, з помилками у визначеннях чи логіці; не може пояснити приклади з практики.

Ф (незадовільно) – не володіє базовими поняттями курсу; не здатен пояснити принципи роботи основних фреймворків або відповісти навіть на допоміжні запитання.

Критерії оцінювання практичних занять

Практичне заняття №1

А (відмінно) – студент створив повністю працездатний застосунок у .NET MAUI; реалізовано інтерфейс з кількома елементами управління; додано логіку обробки подій; проєкт структурований і не містить помилок.

В (дуже добре) – застосунок створено і працює стабільно; реалізовано основні елементи інтерфейсу та події; є незначні недоліки в структурі або стилі коду.

С (добре) – основна функціональність застосунку реалізована; проєкт компілюється і запускається; інтерфейс спрощений або має окремі недоліки.

Д (задовільно) – студент частково реалізував застосунок; присутні помилки або неповна логіка, але загальна структура .NET MAUI зрозуміла.

Е (достатньо для зарахування) – є мінімальна реалізація: створено базовий проєкт, який компілюється, але не має функціональності чи оформлення.

Ф (незадовільно) – робота відсутня, не запускається або не відповідає вимогам завдання.

Практичне заняття №2. Створення додатку на Flutter

А (відмінно) – студент упевнено працює у середовищі Flutter; створив повнофункціональний застосунок із коректним інтерфейсом; використовує віджети різних типів; логіка програми реалізована без помилок.

В (дуже добре) – застосунок створено і працює; інтерфейс оформлений грамотно, але має незначні недоліки; логіка реалізована частково або з мінімальними помилками.

С (добре) – основна функціональність реалізована; застосунок запускається; структура коду зрозуміла, але відсутні деякі елементи інтерфейсу чи оптимізації.

Д (задовільно) – застосунок частково працює або не має завершеного інтерфейсу; присутні помилки, але студент розуміє основні принципи Flutter.

Е (достатньо для зарахування) – здійснено спробу створити застосунок; мінімальний код працює, але відсутня основна логіка чи інтерфейс.

Ф (незадовільно) – робота не виконана або не запускається; студент не демонструє розуміння принципів Flutter.

Практичне заняття №3. Розробка додатків на Ionic Angular

А (відмінно) – студент створив повноцінний додаток з кількома сторінками; реалізовано навігацію, базову логіку та стилізацію; код структурований відповідно до Angular-практик.

В (дуже добре) – реалізовано основні функції додатку та навігацію; є дрібні помилки у логіці або верстці.

С (добре) – додаток працює, але має спрощений інтерфейс; реалізовано лише базові можливості Ionic; відсутня частина функціоналу.

Д (задовільно) – є робочий код, проте з помилками або неповною реалізацією; студент розуміє принципи, але не може їх повністю застосувати.

Е (достатньо для зарахування) – є спроба створити застосунок; мінімальний код Ionic Angular працює частково.

Ф (незадовільно) – завдання не виконане або не запускається; студент не орієнтується у структурі Ionic Angular.

Практичне заняття №4. Робота з Kotlin Multiplatform

А (відмінно) - студент створив працездатний багатоплатформний проект; правильно налаштовано модулі (shared, Android, iOS); реалізовано спільну логіку; продемонстровано розуміння архітектури.

В (дуже добре) – проект створено і працює; основна спільна логіка реалізована; незначні помилки в налаштуванні чи структурі.

С (добре) – проект частково реалізований; код компілюється для однієї платформи; базова структура зрозуміла, але без повної інтеграції.

Д (задовільно) – є спроба створення проекту; студент орієнтується у середовищі, але не реалізував функціональний результат.

Е (достатньо для зарахування) – мінімальні результати: базовий код створено, але не працює як багатоплатформний.

Ф (незадовільно) – робота відсутня або повністю некоректна.

Критерії оцінювання самостійної роботи

А (відмінно) – завдання виконано повністю і коректно; студент продемонстрував глибоке розуміння теми; робота логічна, структурована, без помилок; показана повна самостійність.

В (дуже добре) – завдання виконано майже повністю; дрібні помилки або неточності; студент демонструє хороше розуміння матеріалу; робота структурована, але можна покращити.

С (добре) – основні частини завдання виконані; є помилки або пропущені елементи; студент показав базове розуміння теми; робота читається, але потребує доопрацювання.

Д (задовільно) – завдання виконано частково; багато помилок або пропущені важливі частини; студент має поверхове розуміння матеріалу; робота потребує суттєвого доопрацювання.

Е (достатньо для зарахування) – виконано мінімальні вимоги; робота містить значні помилки; студент демонструє дуже обмежене розуміння теми; робота не структурована.

Ф (незадовільно) – завдання не виконано або виконано некоректно; студент не демонструє розуміння теми; робота відсутня або непридатна для оцінки.

Критерії оцінювання тренінгу

А (відмінно) – студент активно бере участь у всіх етапах тренінгу: постановка завдання, планування, реалізація, тестування, презентація. Виконує усі завдання групи та індивідуально без помилок. Демонструє глибоке розуміння технологій (Flutter, .NET MAUI, Ionic, Kotlin Multiplatform тощо) та принципів кросплатформної розробки. Пропонує оптимальні рішення, аргументує свій вибір технологій. Робота структурована, код / матеріали читабельні та коректні. Презентація результатів логічна, послідовна, відповіді на питання повні та аргументовані. Активно взаємодіє з групою, допомагає однокурсникам.

В (дуже добре) – студент бере участь у більшості етапів тренінгу. Виконує більшість завдань групи та індивідуальних завдань, є дрібні помилки. Демонструє добре розуміння технологій, але деякі пояснення не повні. Робота структурована, невеликі недоліки у кодї чи оформленні. Презентація зрозуміла, аргументація логічна, але не повна. Взаємодія з групою переважно ефективна, інколи потребує допомоги.

С (добре) – студент бере участь у тренінгу частково, виконує лише основні завдання. Демонструє базове розуміння технологій, але відповіді неповні або з помилками. Робота функціональна, є помилки або пропущені частини. Презентація коротка, аргументація частково логічна. Участь у груповій роботі обмежена, активність низька.

D (задовільно) – студент виконує лише частину завдань, робота неповна або з великою кількістю помилок. Розуміння технологій поверхове; аргументація слабка. Код/матеріали неструктуровані, функціональність обмежена. Презентація слабка, відповіді на запитання частково невірні. Участь у групі мінімальна.

E (достатньо для зарахування) – виконано мінімальні завдання; робота має значні помилки. Розуміння технологій дуже обмежене. Код/матеріали частково відсутні або непридатні. Презентація обмежена, логіка та аргументація практично відсутні. Участь у групі слабка, студент майже не долучався.

F (незадовільно) – завдання не виконано або виконано некоректно. Студент не демонструє розуміння технологій або процесу кросплатформної розробки. Робота відсутня або непридатна для оцінки. Відсутня участь у груповій роботі та презентації.

Шкала оцінювання:

За шкалою Університету	За національною шкалою	За шкалою ECTS
90-100	відмінно	A (відмінно)
85-89	добре	B (дуже добре)
75-84		C (добре)
65-74	задовільно	D (задовільно)
60-64		E (достатньо)
35-59	незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)

11. Інструменти, обладнання та програмне забезпечення, використання яких передбачає дисципліна «Інженерія кросплатформного програмного забезпечення»

№	Найменування	Номер теми
1	Мультимедійний проектор	1-12
2	Проекційний екран	1-12
3	Комунікаційне програмне забезпечення (Google Chrome, Firefox)	1-12
4	Операційна система Windows, наявність доступу до мережі Internet	1-12
5	Персональні комп'ютери, ноутбук.	1-12
6	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності)	1-12
7	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-12
8	Базове програмне забезпечення Microsoft Office	1-12
9	Спеціалізоване програмне забезпечення (Visual Studio)	1-12

РЕКОМЕНДОВАНИ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Carl Rippon. ASP.NET Core 5 and React: Full-stack web development using .NET 5, React 17, and TypeScript 4, 2nd Edition. – Packt Publishing Ltd, 2021. – 568 p.
2. Matt Lacey, Marcel Alexander Wagner. Creating Cross-Platform C# Applications with Uno Platform: Build apps with C# and XAML that run on Windows, macOS, iOS, Android, and WebAssembly. – Packt Publishing Ltd, 2021. - 258 p.3
3. Nibedit Dey. Cross-Platform Development with Qt 6 and Modern C++: Design and build applications with modern graphical user interfaces without worrying about platform dependency. – Packt Publishing Ltd, 2021. – 442 p.
4. Bilgin, C. Mobile Development with .NET: Build Cross-platform Mobile Applications with Xamarin.Forms 5 and ASP.NET Core 5, 2nd Edition. Packt Publishing. 2021. – 572 p.
5. Valerio De Sanctis. ASP.NET Core 5 and Angular: Full-stack web development with .NET 5 and Angular 11, 4th Edition. – Packt Publishing Ltd, 2021. – 746 p.
6. <https://dotnet.microsoft.com/en-us/learn/maui/first-app-tutorial/intro>
7. <https://flutter.dev/>
8. <https://reactnative.dev/>
9. <https://ionicframework.com/>
10. <https://www.jetbrains.com/kotlin-multiplatform/>