

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчальної роботи

комп'ютерних наук

кібернетики

Олена КІШПУР

«12»

2021 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
WEB-ТЕХНОЛОГІЇ

для студентів

галузь знань **12 «Інформаційні технології»**  
(шифр і назва)

спеціальність **122 «Комп'ютерні науки»**  
(шифр і назва спеціальності)

освітній рівень **бакалавр**  
(молодший бакалавр, бакалавр, магістр)

освітня програма **«Інформатика»**  
(назва освітньої програми)

вид дисципліни **вибіркова**

вибірковий блок **«Теорія та технології програмування»**

Форма навчання

денна

Навчальний рік

2021/2022

Семестр

5

Кількість кредитів ECTS

3

Мова викладання, навчання

та оцінювання

українська

Форма заключного контролю

залік

Викладачі: к.ф.-м.н., доц. Панченко Т.В. (лекції)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
(підпис, ПІБ, дата)

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.

(підпис, ПІБ, дата)

КИЇВ – 2021

Розробник: Панченко Т.В., к.ф.-м.н., доцент кафедри «Теорії та технології програмування»



ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»

  
\_\_\_\_\_ (підпис)

Микола НІКІТЧЕНКО  
(прізвище та ініціали)

Протокол № 6 від «11» лютого 2021 р.

Схвалено гарантом освітньо-професійної програми «Інформатика»

  
\_\_\_\_\_ (підпис)

Людмила ОМЕЛЬЧУК «11» лютого 2021 рік

(прізвище та ініціали)

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «11» лютого 2021 року № 7

Голова науково-методичної комісії \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Людмила ОМЕЛЬЧУК

**1. Мета дисципліни** – опанування базових знань з сучасних Web-технологій та основ Web-програмування, включаючи Web-Frameworks.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):**

1. *Знати:* основні поняття з основ баз даних, принципів проектування систем, основи програмування.
2. *Вміти:* аналізувати вимоги щодо проектування систем та формалізувати їх, шукати і знаходити потрібну інформацію.
3. *Володіти елементарними навичками:* з дискретної математики, математичної логіки та основ програмування.

**3. Анотація навчальної дисципліни:**

Навчальна дисципліна «WEB-технології» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, освітньо-професійної програми – „Інформатика”.

Дана дисципліна є вибірковою навчальною дисципліною за *програмою “Інформатика”*.

Викладається в 5 семестрі 3 курсу бакалаврату в обсязі 90 годин.

(3 кредити ECTS) зокрема: *лекції – 38 год., самостійна робота – 50 год., консультації – 2 год.* У курсі передбачено 2 частини та 2 контрольні роботи. Завершується дисципліна – **заліком в 5 семестрі.**

В результаті вивчення навчальної дисципліни студент повинен:

**знати:** принципи проектування Web-систем, основні протоколи та методи взаємодії, технології та мови XML, JSON, AJAX, SOA та Web-сервіси, MVC Web Framework, ORM, користуватись та розробляти API, основи налаштування та адміністрування у Web-системах, особливості HTML5;

**вміти:** проектувати та розробляти Web-базовані клієнт-серверні системи, оптимізувати їх швидкодію, налаштовувати роботу з базою даних, використовувати кеш, програмувати Javascript, jQuery, AJAX.

**4. Завдання (навчальні цілі):**

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно освітньої кваліфікації «Бакалавр з комп’ютерних наук».

Зокрема:

- здатність розробляти й управляти проектами;
- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктноорієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, сховища даних і бази знань, для забезпечення обчислювальних потреб багатьох користувачів, обробки транзакцій, у тому числі на хмарних сервісах;
- здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення;
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

## 5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	Знати принципи проектування Web-систем	Лекція, лабораторні заняття	Контрольна робота 1 60% правильних відповідей, лабораторна робота	10%
PH1.2	Знати технології та мови XML, JSON, AJAX, SOA та Web-сервіси, MVC Web Framework, ORM, користуватись та розробляти API	Лекція, лабораторні заняття	Контрольна робота 1 60% правильних відповідей, лабораторна робота	15%
PH1.3	Знати основи налаштування та адміністрування у Web-системах, особливості HTML5	Лекція, лабораторні заняття	Контрольна робота 2 60% правильних відповідей, лабораторна робота	25%
PH2.1	Вміти проектувати та розробляти Web-базовані клієнт-серверні системи, налаштовувати роботу з базою даних, використовувати кеш, програмувати Javascript, jQuery, AJAX.	Лекція, лабораторні заняття, самостійна робота	Лабораторна робота	30%
PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування, розробки специфікацій та програм.	Лекція, лабораторні заняття	Лабораторна робота	10%
PH4.1	Організувати свою самостійну роботу для досягнення результату.	Самостійна робота	Лабораторна робота	10%

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	PH 1.1	PH 1.2	PH 1.3	PH 2.1	PH 3.1	PH 4.1
<b>Програмні результати навчання</b> (з опису освітньої програми)						
<b>ПРН18.2.</b> Аналізувати, оцінювати і вибрати інструментальні та обчислювальні засоби, парадигми, технології, алгоритмічні і програмні рішення при проектуванні та розробці програмних систем.	+	+	+	+	+	+
<b>ПРН20.2.</b> Мотивовано обирати технології програмування для розв'язання завдань створення і супроводження програмного забезпечення.				+	+	+

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### - семестрове оцінювання:

1. *Контрольна робота 1: РН 1.1., РН 1.2,— 20 балів/12 балів.*
2. *Контрольна робота 2: РН1.3 - 20 балів/12 балів.*
3. *Лабораторна робота: РН1.2, РН1.3, РН 2.1, РН3.1, РН4.1 - 60 балів/36 балів.*

- **підсумкове оцінювання у формі заліку.** *Виставляється за результатами роботи студентами впродовж усього семестру та не передбачає додаткових заходів оцінювання для успішних студентів.*

### 7.1 Організація оцінювання:

#### Терміни проведення форм оцінювання:

1. *Контрольна робота 1: до 5 тижня семестру.*
2. *Контрольна робота 2: до 12 тижня семестру.*
3. *Лабораторна робота: до 14 тижня семестру.*

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

Контрольна робота 1, 2 складається з чотирьох теоретичних питань за темами лекційних занять. Кожне питання оцінюється максимально на 5 балів.

#### Завдання для лабораторної роботи.

1. **MVC.** Розробити систему, використовуючи MVC-підхід (в моделі 2+ сутностей зі зв'язками, реалізувати CRUD) та викласти код в GitHub
2. **Cloud.** Викласти проект в Інтернет (Azure або інше cloud-сховище)
3. **JSON API.** Додати до проекту реалізацію CRUD-функцій через JSON (без класичних View: GET-запити вертають множину JSON-об'єктів, POST-запити - з тілом JSON зі змінами - вертають "Ok"/"Error")
4. **Bootstrap Landing.** Розробити Landing-page об'єктів предметної області, використовуючи Bootstrap (вся інформація - на одній сторінці зі scroll, контакти постійно на екрані)
  - 4.1. \* зробити SEO-оптимізацію сторінки
5. **API.** Скористатись Google / MapBox / OpenStreetMap / ... Map сервісом для відображення об'єктів на мапі.
  - 5.1. \* додати (статистичні) міні-діаграми/графіки на мапу
  - 5.2. \* інтегрувати взаємодію з іншим API: Privat24 / WebHooks : Telegram (Bot) / ...
6. **JavaScript + JQuery (+AJAX). Autocomplete.** Drop-Down елемент(и) в MVC-проекті замінити на autocomplete selector (використати JQuery UI Autocomplete, SELECT2 або інший елемент) із завантаженням підходящих елементів (що містять введений <input> рядок) від 3 символів із невеликою затримкою після вводу з клавіатури у це поле.
7. **Connectivity. Web Sockets.** Реалізувати клієнт-серверну взаємодію за допомогою Web Sockets, Long Poll, Adaptive Polls (+ інші) на прикладі проекту (окрім чат-клієнта!) та провести порівняльний аналіз різних способів взаємодії (ефективність використання ресурсів, навантаження на мережу, накладні витрати, тощо).
8. **2-do list (Dynamic UI).** Реалізувати динамічний ToDo-List: додавати, переглядати список. При закритті-відкритті браузера - список зберігати та відновлювати (автоматично).
  - 8.1. \* сортування елементів (drag-n-drop зі збереженням порядку)

8.2. \* при відкритті списку з 2 браузерів (2 пристроїв) - синхронізувати їх: зміни на одному (майже) миттєво відображаються на іншому пристрої

8.3. \*\* зміни з 8.2. - візуалізуються у динаміці (рух, підсвітка, тощо)

9. **React Front-End.** Розробка проекту на React (або іншому Front-End Framework) для JSON API розробленого у 3 задачі Web-проекту.

## 7.2 Шкала відповідності оцінок

<b>Зараховано / Passed</b>	60-100
<b>Не зараховано / Fail</b>	0-59

## 8. Структура навчальної дисципліни. Тематичний план лекцій

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаб. заняття	Сам. р-та
	<b>Частина 1. Основи Web-програмування. Back-End. Web-сервіси.</b>			
1	<b>Тема 1.</b> Internet. Structure: Hosts, Autonomous Systems. ICANN, IANA, W3C, IEEE. RFC. TCP/IP. Sockets. IP Routing. DNS. Web-Server. Cloud Technologies. Server & DNS Remote Management. HTTP. Client-Server Architecture. Request - Response Cycle. HTTP Headers. Cache management. MIME Types. Encodings. Cookies. Sessions. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	4		2
2	<b>Тема 2.</b> MVC (from Routing to Rendering). ORM. Code First / DB First. SQL vs. ORM. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		6
3	<b>Тема 3.</b> Web Project Deployment & Publication. Browsing & Developer Console. Request Tracking. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		2
4	<b>Тема 4.</b> API. XML Technologies. Web Services. SOAP & WSDL. Rest API. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		6
5	<b>Тема 5.</b> Performance. Distributed Systems. Load Balancing. High Loading. Load Testing. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		2
6	<b>Тема 6.</b> Web Architecture. Micro Services. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		2
7	<b>Тема 7.</b> Security. Authentication & Authorization. SSL & TLS. Penetration Test. Tools (SSL Labs, Accunetix, Microsoft Network Monitor). PKI. Ciphers. Digital Signature. XSS. CRDF. OWASP. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		4
	<i>Контрольна робота 1</i>	2		
	<b>Частина 2. Front-End. Практики Web-розробки.</b>			
8	<b>Тема 8.</b> HTML. DOM. CSS. Bootstrap. UI & UX. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	4		4
9	<b>Тема 9.</b> JavaScript Basics. Syntax & DOM - manipulating documents. JSON. AJAX. JQuery. Cross-browser compatibility. JQuery UI. Events & Handlers. Event object. Callbacks. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	4		4
10	<b>Тема 10.</b> JavaScript Features. Web APIs. Navigation & other Browser APIs. Third-party APIs. Graphics (2D, 3D). Closures. Classes & Objects (OOP). The scope and “this” keyword. Threads. Asynchronous JavaScript: Promises, async+await. Timers (timeout, interval).	2		4

	<i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>			
11	<b>Тема 11.</b> Search Engines. SEO. User Behavior Analytics (Tracking Systems). <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		4
12	<b>Тема 12.</b> React Framework. Front-End Project Development. Style Guides. Best Practices. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		4
13	<b>Тема 13.</b> Application State Management. Routing. Internationalization. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		4
14	<b>Тема 14.</b> Version Control Systems. GitHub. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	2		2
<i>Контрольна робота 2</i>		2		
<b>ВСЬОГО</b>		38		50

**Загальний обсяг 90 год.**, в тому числі:

Лекцій – 38 год.

Самостійна робота - 50год.

Консультації – 2 год.

## 9. Рекомендовані джерела:

### **Основна:**

1. Request For Comment (RFC). (<http://www.faqs.org/rfc/>)
2. WWW Consortium (W3C) Official Page. (<http://www.w3c.org/>)
3. Hyper Text Markup Language (HTML) Standard. (<http://www.w3c.org/MarkUp/>)
4. Cascade Style Sheets (CSS) Standard. (<http://www.w3c.org/Style/CSS/>)
5. eXtensible Markup Language (XML) Standard. (<http://www.w3c.org/XML/>)
6. Document Object Model (DOM) Standard. (<http://www.w3c.org/DOM/>)
7. JavaScript Specification. (<https://www.ecma-international.org/publications/standards/Ecma-262.htm>)
8. Learning JavaScript.  
([https://developer.mozilla.org/uk/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics))
9. React. (<https://uk.reactjs.org/>)
10. General Problem Solver. (<https://www.google.com>)