

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

« ____ » _____ 2019 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**

Форма навчання	денна
Навчальний рік	2019/2020
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 – 2019

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

ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»

_____ Нікітченко М.С.
(підпис) (прізвище та ініціали)

Протокол № ____ від « ____ » _____ 2019 р.

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

Протокол від « ____ » _____ 2019 року № ____

Голова науково-методичної комісії _____ Омельчук Л.Л.
(підпис) (прізвище та ініціали)

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від « ____ » _____ 2019 року № ____

Голова вченої ради факультету _____ А.В. Анісімов

ВСТУП

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 – 28 год., лабораторних занять – 12 год., самостійна робота – 50 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’єктно-орієнтованого, 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, практичні заняття	Контрольна робота 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, практичні заняття	Контрольна робота 60% правильних відповідей, лабораторна робота	30%
PH1.3	Знати основи налаштування та адміністрування у Web-системах, особливості HTML5	Лекція, практичні заняття	Контрольна робота 60% правильних відповідей, лабораторна робота	10%
PH2.1	Вміти проектувати та розробляти Web-базовані клієнт-серверні системи, налаштовувати роботу з базою даних, використовувати кеши, програмувати Javascript, jQuery, AJAX.	Лекція, практичні заняття, самостійна робота	Поточне оцінювання, лабораторна робота	30%
PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування, розробки специфікацій та програм.	Лекція, практичні заняття	Поточне оцінювання, лабораторна робота	10%
PH4.1	Організовувати свою самостійну роботу для досягнення результату.	Самостійна робота	Поточне оцінювання, 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
Програмні результати навчання						
(з опису освітньої програми)						
ПРН18.2. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, парадигми, технології, алгоритмічні і програмні рішення при проектуванні та розробці програмних систем.	+	+	+	+	+	

ПРН20.2. 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/10 балів.
2. Контрольна робота 2: РН1.3 - 20 балів/10 балів.
3. Лабораторна робота: РН 2.1, РН3.1, РН4.1 - 60 балів/40 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.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

7.2 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій

№ лекції	Назва лекції	Кількість годин		
		Лекції	Практ занять	Сам. р-та
	Частина 1. Основи Web-програмування. Back-End. Web-сервіси.			
1	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.	2		2
2	MVC (from Routing to Rendering). ORM. Code First / DB First. SQL vs. ORM.	2	2	6
3	Web Project Deployment & Publication. Browsing & Developer Console. Request Tracking.	2		2
4	API. XML Technologies. Web Services. SOAP & WSDL. Rest API.	2	2	6
5	Performance. Distributed Systems. Load Balancing. High Loading. Load Testing.	2		2
6	Web Architecture. Micro Services.	2	2	2
7	Security. Authentication & Authorization. SSL & TLS. Penetration Test. Tools (SSL Labs, Accunetix, Microsoft Network Monitor). PKI. Ciphers. Digital Signature. XSS. CRDF. OWASP.	2		2
<i>Контрольна робота 1</i>				2
Всього по частині 1		14	6	24
	Частина 2. Front-End. Практику Web-розробки.			
8	HTML. DOM. CSS. Bootstrap. UI & UX.	2	2	4
9	JavaScript Basics. Syntax & DOM - manipulating documents. JSON. AJAX. JQuery. Cross-browser compatibility. JQuery UI. Events & Handlers. Event object. Callbacks.	2		4
10	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	2	4
11	Search Engines. SEO. User Behavior Analytics (Tracking Systems).	2		2
12	React Framework. Front-End Project Development. Style Guides. Best Practices.	2	2	4
13	Application State Management. Routing. Internationalization.	2		4
14	Version Control Systems. GitHub.	2		2
<i>Контрольна робота 2</i>				2
Всього по частині 2		14	6	26
ВСЬОГО		28	12	50

Загальний обсяг 90 год., в тому числі:

Лекцій – 28 год.

Практичних занять – 12 год

Самостійна робота - 50 год.

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.

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)
9. React. (<https://uk.reactjs.org/>)
10. General Problem Solver. (<https://www.google.com>)