


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» <i>(шифр і назва)</i>
спеціальність	122 «Комп'ютерні науки» <i>(шифр і назва спеціальності)</i>
освітній рівень	бакалавр <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	«Інформатика» <i>(назва освітньої програми)</i>
вид дисципліни	вибіркова
вибірковий блок	«Теорія та технології програмування»

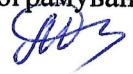
Форма навчання	заочна
Навчальний рік	2019/2020
Семестр	6
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Пролонговано: на 2019/2021 н.р.  «02» 2020 р.

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»



(підпис)

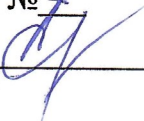
Нікітченко М.С.
(прізвище та ініціали)

Протокол № 1 від «28» 08 2019 р.

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

Протокол від «28» 08 2019 року № 1

Голова науково-методичної комісії



(підпис)

Омельчук Л.Л.
(прізвище та ініціали)

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 – 5 год., лабораторних занять – 2 год., самостійна робота – 82 год., консультації – 1 год.* У курсі передбачено **2 частини**. Завершується дисципліна – **заліком в 6 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			
РН1.1	Знати принципи проектування Web-систем	Лекція, лабораторні заняття, самостійна робота	Контрольна робота 1 60% правильних відповідей, лабораторна робота	10%
РН1.2	Знати технології та мови XML, JSON, AJAX, SOA та Web-сервіси, MVC Web Framework, ORM, користуватись та розробляти API	Лекція, лабораторні заняття, самостійна робота	Контрольна робота 1 60% правильних відповідей, лабораторна робота	15%
РН1.3	Знати основи налаштування та адміністрування у Web-системах, особливості HTML5	Лекція, лабораторні заняття, самостійна робота	Контрольна робота 2 60% правильних відповідей, лабораторна робота	25%
РН2.1	Вміти проектувати та розробляти Web-базовані клієнт-серверні системи, налаштовувати роботу з базою даних, використовувати кеши, програмувати Javascript, jQuery, AJAX.	Лекція, лабораторні заняття, самостійна робота	Лабораторна робота	30%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування, розробки специфікацій та програм.	Лекція, лабораторні заняття, самостійна робота	Лабораторна робота	10%
РН4.1	Організувати свою самостійну роботу для досягнення результату.	Самостійна робота	Лабораторна робота	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 3.1	РН 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/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 третього лекційного заняття.
2. Контрольна робота 2: до третього лекційного заняття.
3. Лабораторна робота: до третього лекційного заняття.

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 (оклім 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

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаб. заняття	Сам. р-та
	Частина 1. Основи Web-програмування. Back-End. Web-сервіси.			
1	Тема 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. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	1		5
2	Тема 2. MVC (from Routing to Rendering). ORM. Code First / DB First. SQL vs. ORM. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	1		5
3	Тема 3. Web Project Deployment & Publication. Browsing & Developer Console. Request Tracking. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	1		5
4	Тема 4. API. XML Technologies. Web Services. SOAP & WSDL. Rest API. <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			5
5	Тема 5. Performance. Distributed Systems. Load Balancing. High Loading. Load Testing. <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			5
6	Тема 6. Web Architecture. Micro Services. <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			5
7	Тема 7. 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 матеріалу, виконання лабораторної роботи.</i>			5
	<i>Контрольна робота 1</i>			5
	Всього по частині 1	3		40
	Частина 2. Front-End. Практики Web-розробки.			
8	Тема 8. HTML. DOM. CSS. Bootstrap. UI & UX. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>	1		5
9	Тема 9. 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>	1		5
10	Тема 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			5

	JavaScript: Promises, async+await. Timers (timeout, interval). <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			
11	Тема 11. Search Engines. SEO. User Behavior Analytics (Tracking Systems). <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			5
12	Тема 12. React Framework. Front-End Project Development. Style Guides. Best Practices. <i>Самостійна робота: опрацювання лекційного матеріалу, виконання лабораторної роботи.</i>		2	5
13	Тема 13. Application State Management. Routing. Internationalization. <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			6
14	Тема 14. Version Control Systems. GitHub. <i>Самостійна робота: опрацювання матеріалу, виконання лабораторної роботи.</i>			6
<i>Контрольна робота 2</i>				5
Всього по частині 2		2	2	42
ВСЬОГО		5	2	82

Загальний обсяг 90 год., в тому числі:

Лекцій – 5 год.

Лабораторних занять – 2 год

Самостійна робота - 82 год.

Консультації – 1 год.

9. Рекомендовані джерела:

Основні:

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>)