

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



Олег САВЕНКО
2023 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Веб-технології

Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Рівень вищої освіти – Перший бакалаврський
Освітньо-професійна програма – Інженерія програмного забезпечення
Обсяг дисципліни – 6 кредитів ЄКТС, **Шифр дисципліни** – ОПП.10
Статус дисципліни: обов'язкова, **Мова навчання** Англійська, українська
Факультет – Інформаційних технологій
Кафедра – Інженерії програмного забезпечення


Форма здобуття освіти	Курс	Семестр	Обсяг дисципліни Кредити ЄКТС	Кількість годин						Форма семестрового контролю			
				Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС	Курсовий проєкт	Курсова робота	Залік	Іспит
				Разом	Лекції	Лабораторні роботи	Практичні заняття						
Очна (денна)	3	5	6	68	34	34		112	+		+		
Разом			6	68	34	34		112	1		1		

Робоча програма складена на основі Стандарту вищої освіти, освітньо-професійної програми підготовки бакалаврів 2023 року та навчального плану


Програма складена  Оксана ЯШИНА

Схвалена на засіданні кафедри інженерії програмного забезпечення

Протокол № 1 від 31.08 2023 р.

Зав.кафедри інженерії програмного забезпечення  Леонід БЕДРАТЮК

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

Голова вченої ради  Олег САВЕНКО

Хмельницький 2023

ВЕБ ТЕХНОЛОГІЇ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська, Англійська
Семестр	П'ятий
Обсяг кредитів ЄКТС	6
Форми здобуття освіти	Очна (денна)

Результати навчання. Студент, який успішно завершив вивчення дисципліни, має: аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

Пререквізити – Об'єктно-орієнтоване програмування

Кореквізити – Архітектура та проектування програмного забезпечення.

Зміст навчальної дисципліни (розділи). Основні поняття веброботки. Методології веброботки. Адресація у мережі Інтернет. Системи керування версіями GIT. Технологія JavaScript. Робота з файлами JSON, XML. Використання CMS. Технології ASP.NET MVC. Робота з контролерами, моделями, представленнями. Маршрутизація. Фільтри та робота з ними. Авторизація та аутентифікація в MVC.

Запланована навчальна діяльність: лекції – 34 год., лабораторні заняття – 34 год., самостійна робота – 112 год., разом – 180 год.

Методи навчання: лекції (з використанням методів проблемного навчання і візуалізації), лабораторні заняття (з використанням комп'ютерних методів та засобів навчання), самостійна робота, курсове проектування.

Форми оцінювання результатів навчання: захист лабораторних робіт, проміжні тестові контролі, захист курсового проекту.

Форма семестрового контролю: іспит.

Навчальні ресурси:

1. WWW Consortium (W3C) Official Page. URL: <http://www.w3c.org/>.
2. ASP.NET Core documentation. URL: <https://learn.microsoft.com/aspnet/core/security/authorization/roles?view=aspnetcore-9.0>
3. Бернерс-Лі Заснування павутини = Weaving the web. The original design and ultimate destiny of the world wide web: З чого починалася і до чого прийде Всесвітня мережа / Тім Бернерс-Лі разом з Марком Фічетті; пер. з англ. А. Іщенко. – Київ : Києво-Могилянська академія, 2020.– 208 с.
4. Цеслів О.В. WEB-програмування: навч. посібник / О.В. Цеслів: М-во освіти і науки, молоді та спорту України, Нац. техн. ун-т України «Київ. політехн. ін-т». – Київ : НТУУ «КПІ», 2020. – 149 с.
5. Rajkumar Buyya, James Broberg, Andrzej Goscinski Cloud computing Principles and Paradigms. John Wiley & Sons, Inc., New Jersey, Canada, 2021. – 637 с.
6. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua>.
7. Електронна бібліотека університету. Доступ до ресурсу: <http://library.khmnu.edu.ua>

Викладач: кандидат технічних наук, доцент Оксана ЯШИНА.

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна «Веб технології» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення».

Пререквізити – Об’єктно-орієнтоване програмування. Людино-машинна взаємодія

Кореквізити – Архітектура та проектування програмного забезпечення.

Мета дисципліни: опанування студентами базових знань з сучасних веб-технологій та основ веб-програмування, включаючи технології вебсервісів, ASP.NET MVC, засвоєння теоретичних і практичних знань із питань перетворення, передавання та використання інформації в глобальному просторі, розкриття значення інформаційних процесів у формуванні сучасної системно-інформаційної схеми інформатизованого світу.

Предмет дисципліни: Предметом вивчення дисципліни є принципи сучасного програмування для інтернет-простору, а також методи їх використання при розробці вебресурсів різноманітного призначення.

Завдання дисципліни:

Надати студентам знання і практичні навички з розробки та реалізації вебпроектів із застосуванням сучасних мов та засобів програмування.

Відповідно до *Стандарту вищої освіти* із та освітньої програми дисципліна сприяє забезпеченню:

компетентностей:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ФК5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

програмних результатів навчання:

ПРН1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки

ПРН6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН15. Мотивовано обирати мови програмування та технології розробки для розв’язання завдань створення і супроводження програмного забезпечення.

Результати навчання. Студент, який успішно завершив вивчення дисципліни, має: аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення. Мотивовано обирати мови програмування та технології розробки для розв’язання завдань створення і супроводження програмного забезпечення.

Політика дисципліни Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов’язаний відвідувати лекції, практичні заняття,

лабораторні роботи, тощо, згідно з розкладом, не запізнюватися на заняття, виконувати усі завдання та контрольні точки відповідно до графіка. Пропущені практичні заняття і лабораторні роботи студент зобов'язаний опрацювати самостійно у повному обсязі і відзвітувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До практичних занять і лабораторних робіт студент має підготуватися за відповідною темою і проявляти активність. Набутті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання та визначення академічної різниці у ХНУ.

4. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:		
	Лекції	Лаб. роботи	СРС (в тому числі КП)
Тема 1. Методології та технології веб-розробки.	6	8	24
Тема 2. Технологія JavaScript.	8	8	24
Тема 3. Технології ASP.NET MVC.	20	16	40
Разом	34	34	112

5.ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу

№ лекції	Перелік тем лекцій, їх анотації	К-сть годин
1	2	3
1	Тема 1. Методології та технології веброзробки. Лекція 1. Основні поняття веброзробки. Функції та задачі розробника. Методології веброзробки. Поняття про вебзастосунки та компоненти для роботи з ними. Сервіс WWW: компоненти, програмне забезпечення. Сучасні веббраузери та принципи їх роботи. Огляд CMS систем, фреймворків, мов програмування, web-серверів і технологій програмування серверних додатків. Класичні та гнучкі методології програмної веброзробки: види, принципи, правила. Інформаційно-довідникові ресурси для вирішення задач веброзробки. [2, 5,7,8]	2
2	Лекція 2. Адресація у мережі Інтернет. Протоколи передачі даних. Протокол HTTP/HTTPS. Методи HTTP-запиту. Архітектура «клієнт-сервер». Передача документів користувача. Літ.: [2, 5,7,8]	2
3	Лекція 3. Системи керування версіями GIT. Основні поняття про системи керування версіями. Встановлення та налаштування GIT. Автоматизація робочих процесів за допомогою GitHub Action. [8,9,16]	2
4	Тема 2. Технологія JavaScript. Лекція 4. Основи JavaScript. Структура коду, типи даних та основні оператори. Підключення JavaScript до вебсторінки різними способами. Змінні, типи даних та перетворення типів. Робота з умовними операторами Основи JavaScript: об'єкти, масиви, цикли та функції. методи об'єктів та масивів. Колекції Map та Set, WeakMap, WeakSet. Робота циклів і обробка даних для написання бізнес-логіки програм. Види функцій та можливості повторного використання коду [2, 5,7,8].	2
5	Лекція 5. JavaScript: класи ES5 і ES6. Прототипне успадкування. Функції конструктора. Класи ES6. ООП в JavaScript: відмінності від інших мов. Ключове слово this та контекст виклику. Основні відмінності ООП JavaScript від ООП в інших мовах. Розбиття коду на незалежні модулі	2
6	Лекція 6. Робота з файлами JSON, XML. Робота з різними форматами мовою програмування JavaScript. Запити, обробка подій та введення користувача. Налаштовування та надсилання AJAX-запитів. Створення DOM-елементів та додавання їх на сторінку. Правильний запит та отримання даних із сервера. Обробка подій користувача Авторизація та аутентифікація в браузерів. Використання регулярних виразів для перевірки правильності вводу інформації користувача. Різні типи авторизацій. Cookie та сесія. Методи зберігання даних у браузері	2
7	Лекція 7. Використання CMS. Історія виникнення. CMS WordPress, Joomla та інші. Відмінності CMS від створених сайтів на розроблених платформах.	2
8	Тема 3. Технології MVC ASP.NET MVC. Лекція 8. Технології ASP.NET MVC. Поняття технології ASP.NET MVC. Контролер (Controller), Модель (Model). Представлення (View). Обробка запитів MVC-застосунком. Підхід до розробки MVC-застосунків. Особливості ASP.NET MVC. Створення першого застосунку. Літ.: [2,3,5,6,7,8].	2

9	Лекція 9. Робота з контролерами. Основи роботи з контролерами. Методи дій та їх параметри. Результати дій. Переадресація та відправлення кодів статусу та помилок. Відправлення файлів в ASP.NET MVC. Контекст запиту HttpContext. Куки. Сесії. Асинхронні методи. Літ.: [3,5,6,7,8].	2
10	Лекція 10. Робота з представленнями. Основи роботи з представленнями. Строго типізовані представлення. Майстер-сторінки. Часткові представлення. HTML-хелпери.Робота з формами. Строго типізовані хелпери. Літ.: [3,5,6,7,8]	2
11	Лекція 11. Робота з моделями. Створення моделей. Підключення до бази даних Шаблонні хелпери. Додавання та видалення моделі. Шаблони формування. Літ.: [3,5,6,7,8]	2
12	Лекція 12. Робота з моделями. Моделі зі складною структурою та робота з ними. Моделі зі зв'язком багато до багатьох. Передача масивів та складних даних в контролер. Літ.: [3,5,6,7,8]	2
13	Лекція 13. Робота з моделями. Міграція баз даних. Створення пагінації. Перевизначення шаблонів формування. Перевизначення шаблонів редагування та зображення. Фільтрація даних. Domain Model та View Model. Керування ієрархічними даними Літ.: [3,5,6,7,8]	2
14	Лекція 14. Маршрутизація. Визначення маршрутів. Робота з маршрутами. Створення обмежень для маршрутів. Генерація вихідних адрес URL. Області. Створення власного обробника маршрутів. Атрибути маршрутизації. Маршрутизація та вкладені ресурси Літ.: [3,5,6,7,8]	2
15	Лекція 15. Метадані та валідація моделі. Анотації даних для відображення властивостей. Основи валідації. Атрибути валідації. Валідація моделі в контролері. Відображення помилок валідації. Створення власної логіки валідації. Літ.: [3,5,6,7,8]	2
16	Лекція 16. Фільтри. Фільтри в ASP.NET MVC. Фільтри аутентифікації. Фільтри авторизації. Валідація моделі в контролері. Відображення помилок валідації. Створення власної логіки валідації. Літ.: [3,5,6,7,8]	2
17	Лекція 17. Авторизація та аутентифікація в MVC. Введення в авторизацію і аутентифікацію. Аутентифікація Windows. Аутентифікація форм. Авторизація в MVC. Налаштування використання SimpleMembershipProvider. Використання універсальних провайдерів. Web Site Administration Tool. Перевизначення провайдера ролей. Провайдер профілів і його перевизначення. Літ.: [3,5,6,7,8]	2
	Разом за семестр	34

5.2. ЗМІСТ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Тема лабораторного заняття	Кількість годин
1	2	3
1	Лабораторна робота №1. Створення простого застосунку в ASP.NET Core MVC. Літ.: [4, 5].	4
2	Лабораторна робота №2. Додавання контролера, представлень та моделей до проєкту. Використання Create, Read, Update та Delete (CRUD). Літ.: [5,11].	4
3	Лабораторна робота №3. Вибір та розробка користувацького інтерфейсу. Літ.: [3,5,6,7].	4
4	Лабораторна робота №4. Робота з базою даних в застосунках. Літ.: [5,6,7, 8, 9, 11].	4
5	Лабораторна робота №5. Додавання пошуку та полів у застосунок ASP.NET Core MVC. [5,6,7].	4
6	Лабораторна робота №6. Авторизація на основі ролей. Літ.: [5,6,7].	4
7	Лабораторна робота №7. Додавання можливостей валідації в застосунок MVC ASP.NET. Літ.: [5,6,7].	4
8	Лабораторна робота №8. Локалізація та мультимовність. Літ.: [5,6,7].	4
9	Підсумкове заняття.	2
	Разом за семестр	34*

Примітка. * по чисельнику – 34 годин, по знаменнику – 32 години

5.3 ЗМІСТ САМОСТІЙНОЇ РОБОТИ

Самостійна робота студентів *денної* форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань, в тому числі курсового проєкту, тощо. Курсовий проєкт виконується згідно із методичними рекомендаціями, що розміщено у модульному середовищі, а також в друкованому варіанті знаходиться на кафедрі ІІЗ.

Номер тижня	Зміст самостійної роботи	Кількість годин
1	2	3
1	Опрацювання лекційного матеріалу, підготовка до виконання лабораторної роботи № 1. Визначення тематики курсового проєкту.	4
2	Підготовка до захисту лабораторної роботи №1 та до виконання лабораторної роботи №2. Визначення завдань до КП.	4
3	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №2. Опрацювання довідкової літератури. Здійснення аналізу предметної області відповідно до тематики КП.	4

4	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 2. Здійснення аналізу існуючих програмних рішень відповідно до тематики КП.	4
5	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 3. Опрацювання довідкової літератури. Розробка ТЗ до КП.	4
6	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 3. Опрацювання довідкової літератури. Проектування структури програми відповідно до тематики КП.	6
7	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 4. Опрацювання довідкової літератури. Проектування архітектури програмної системи відповідно до тематики ПКП.	6
8	Опрацювання лекційного матеріалу та довідкової літератури. Підготовка до ТК1. Проектування бази даних у програмі відповідно до тематики КП.	6
9	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 4. Вибір засобів та технологій реалізації програми відповідно до тематики КП.	6
10	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 5. Опрацювання довідкової літератури. Реалізація програми відповідно до тематики КП.	6
11	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №6. Опрацювання довідкової літератури. Реалізація програми відповідно до тематики КП.	6
12	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 6. Опрацювання довідкової літератури. Реалізація програми відповідно до тематики КП.	6
13	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 7. Опрацювання довідкової літератури. Написання інструкції користувача.	6
14	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 7. Опрацювання довідкової літератури. Тестування програми КП.	6
15	Опрацювання лекційного матеріалу. Виконання лабораторної роботи №8. Опрацювання довідкової літератури. Підготовка до перевірки та надсилання КП викладачу.	6
16	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №8. Опрацювання довідкової літератури. Оформлення пояснювальної записки КП та перевірка на антиплагіат.	6
17	Опрацювання лекційного матеріалу. Підготовка до ТК2. Підготовка до захисту КП.	6
Разом		112

6. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів. Зокрема, лекції проводяться в основному словесними методами, а лабораторні заняття проводяться з використанням інформаційних технологій, майстер-класів, практикумів і мають за мету – набуття студентами практичних навичок з створення програмного продукту за сучасними методиками.

7. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі заліку. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Використовуються наступні методи поточного контролю:

- усне опитування студентів перед допуском до лабораторного заняття;
- допуск до захисту лабораторних робіт;
- захист лабораторних робіт;
- тестування та перевірка працездатності створеного у ході виконання лабораторної роботи програмного, технічного та іншого забезпечення;

Підсумкова оцінка виставляється з урахуванням коефіцієнтів вагомості.

Студент, який не допущений до захисту або не захистив лабораторні роботи, вважається невстигаючим.

Процес оцінювання підготовленості студента можна розділити на етапи:

Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань при проектуванні виробів із шкіри, вміє розширити їх.

Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: відтворювати формулювання визначень поняття мобільних платформ; орієнтуватись в методиках розробки програмного продукту; знати теоретичні основи створення додатків для мобільних платформ.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна здійснювати смислове виділення, пояснення вибору конструктивних рішень щодо створення програмного продукту, методик їх проектування, їх обґрунтування на основі аналізу існуючих. Може чітко визначити структуру програм при їх проектуванні в залежності від їх використання, тобто здатна перенести раніше засвоєнні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєнні знання на нетипові, нестандартні ситуації. Тобто на цьому рівні студент повинен на основі теоретичних знань вміти проектувати різні види програмного продукту за найбільш поширеними методиками, вносити свої пропозиції щодо удосконалення способів проектування та одержання додатків для мобільних платформ; створювати конкурентоспроможні програмні додатки на світовому ринку.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема:

- допуск до виконання лабораторної роботи здійснюється на її початку усним опитуванням студента;
- якість виконання лабораторної роботи оцінюється шляхом перевірки виконаної лабораторної роботи у робочому файлі та в робочому зошиті;
- набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи;

-засвоєння теоретичного матеріалу перевіряється тестовим контролем.

Оцінка, яка виставляється за лабораторне заняття, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення результатів лабораторної роботи у зошиті та в робочому файлі; вміння студента обґрунтовувати прийняті рішення та робити висновки; своєчасний захист лабораторної роботи.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті.

Пропущене з поважної причини лабораторне заняття студент повинен відпрацювати в установленний викладачем термін.

Кожний вид роботи оцінюється за чотирибальною шкалою. Підсумкова оцінка з дисципліни виставляється як середньозважена з усіх видів робіт (табл.1).

Таблиця 1. Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота		Самостійна, індивідуальна робота		Семестровий контроль	
<i>П'ятий семестр</i>					
Лабораторні роботи №:			Тестовий контроль:		Іспит
1-8			ТК 1-2		1
ВК*: 0,4			0,2		0,4
<i>П'ятий семестр (курсовий проект)</i>					
<i>Контрольні точки №</i>					Захист КП
1 розділ	2 розділ	3 розділ	4 розділ	Програма	1
0,1	0,1	0,1	0,1	0,3	0,3

Умовні позначення: ТК – тестовий контроль; ВК – ваговий коефіцієнт.

Оцінка роботи студента за традиційною системою

Оцінка "*відмінно*" виставляється студентові, який вільно володіє матеріалом, чітко, грамотно, логічно, повно і послідовно висвітлює питання, демонструє знання не лише лекційного матеріалу, але й результати самостійного опрацювання літератури, вміння висвітлювати власні міркування та робити висновки.

Найвища оцінка ставиться студенту, який показав всебічне та глибоке знання навчально-програмного матеріалу, розбирається в поняттях, принципах та методах аналізу вимог до ПЗ, а також: який засвоїв основну і знайомий з додатковою літературою, засвоїв основні поняття дисципліни; дав повні і точні відповіді на всі запитання з поясненнями і висновками; проявив творчі здібності в розумінні, викладенні і використанні навчального матеріалу.

Оцінка "*добре*" виставляється студентові, який добре володіє навчальним матеріалом, грамотно і логічно викладає його в усній чи письмовій формі, допускаючи при цьому не більше одного-двох недоліків чи незначних помилок, Обмежується викладом основних теоретичних положень, не володіє матеріалом додаткової літератури, хоча загалом має тверді знання.

"*Задовільно*" отримує студент, що показав знання основного матеріалу в обсязі, необхідному для подальшого навчання (основні поняття, принципи, закономірності), а також: знайомий з основною літературою; при відповідях на запитання відчуває себе невпевнено, допускає незначні помилки; при допусканні погрешностей у відповідях володіє знаннями для їх виправлення під керівництвом викладача.

Оцінка "*незадовільно*" виставляється студентові, який демонструє не зовсім повні знання курсу навчальної дисципліни чи її окремої теми, допускає при викладанні навчального матеріалу суттєві помилки, не орієнтується у лекційному матеріалі та матеріалі основної і додаткової літератури.

"Незадовільно" ставиться студентові, який має пробіли в знаннях основного матеріалу, а також: допустив принципові помилки, не розуміє основних концептуальних положень дисципліни; на запитання не дає задовільних відповідей, не справляється з рішенням задач; не може продовжувати навчання без додаткових занять з дисципліни.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю. У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у табл. 2.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Таблиця 2. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Бали	Вітчизняна система	
A	4,75-5,00	5	ВІДМІННО - глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
B	4,25-4,74	4	ДОБРЕ - повне знання навчального матеріалу з кількома незначними помилками
C	3,75-4,24	4	ДОБРЕ - в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25-3,74	3	ЗАДОВІЛЬНО - неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00-3,24	3	ЗАДОВІЛЬНО - неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	НЕЗАДОВІЛЬНО - безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2	НЕЗАДОВІЛЬНО - необхідна серйозна подальша робота і повторне вивчення дисципліни

8. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ РЕЗУЛЬТАТІВ НАВЧАННЯ

1. Що таке вебзастосунок?
2. Що таке браузер?
3. Опишіть цикл обробки запиту до веб-додатку від клієнта.
4. Для чого необхідні технології розробки веб-додатків (такі як ASP.NET, PHP, Ruby On Rails та ін.)
5. Як працює протокол HTTP і для чого він потрібен?
6. Що таке заголовки HTTP-повідомлення і для чого вони потрібні?
7. Що таке тіло HTTP-повідомлення?
8. Яким чином в HTTP-повідомленні заголовки відокремлюються від тіла повідомлення?
9. Що таке метод HTTP-запиту?
10. Що таке статусний код HTTP-відповіді?

11. Наведіть приклади HTTP-заголовків HTTP-запиту і HTTP-відповіді.
12. Як працює захищений протокол HTTPS?
13. Що таке вебсервер?
14. На основі яких інтерфейсів може взаємодіяти вебсервер і вебзастосунок?
15. Назвіть найбільш популярні реалізації веб-серверів.
16. В межах якого вебсервера працюють програми ASP.NET?
17. Назвіть етапи розробки MVC-додатків.
18. Опишіть життєвий цикл запиту для MVCзастосунку.
19. Microsoft ASP.NET MVC це – ?
20. Що таке Контролер?
21. Що таке Модель?
22. Що таке Представлення?
23. Алгоритм роботи MVC додатку.
24. Для вирішення яких задач призначена платформа ASP.NET MVC?
25. Основи контролерів.
26. Визначення інтерфейсу IController.
27. Методи дій.
28. Отримання вхідних даних. Передача параметрів.
29. Класи, похідні від ActionResult. 60. Передача даних з контролера в представлення.
61. Переадресація. Об'єкти RedirectResult, RedirectPermanent.
62. Метод RedirectToAction.
63. Об'єкт ContentResult.
64. Контекст запиту.
65. Кукі.
66. Введення в представлення.
67. Представлення з використанням синтаксису Razor.
68. Шляхи до файлів представлень.
69. Типи шаблонів представлення.
70. Строго типізовані представлення.
71. Движок представлень Razor.
72. Майстер-сторінки.
73. Сторінка _ViewStart.
74. Часткові представлення.
75. Хелпер Html.Partial.
76. 77. Хелпер Html.RenderPartial.
78. HTML-хелпери.
79. Строго типізовані хелпери. Лямбда-вирази.
80. Моделі і бази даних.
81. Фреймворк Entity Framework.
82. Підключення до бази даних.
83. Шаблонні хелпери.
84. Редагування моделі.
85. Шаблони формування.
86. Моделі зі складною структурою.
87. Робота зі складними моделями.
88. Передача даних в контролер.
89. Маршрутизація.
90. Визначення маршрутів.
91. Робота з маршрутами.
92. Створення обмежень для маршрутів.
93. Генерація вихідних адрес URL.
94. Області (Areas).
95. Створення власного обробника маршрутів.

96. Метадані.
97. Валідація моделі.
98. Анотації даних для відображення властивостей.
99. Атрибути валідації.
100. Валідація моделі в контролері.
101. Відображення помилок валідації.
102. Створення власної логіки валідації.
103. Фільтри.
104. Фільтри авторизації.
105. Фільтри виключень.
106. Фільтри дій і результатів.
107. Глобальні фільтри.
108. Вбудовані фільтри.
109. Прив'язка моделі.
110. Створення прив'язчика моделі.
111. jQuery і AJAX.
112. Використання JavaScript/jQuery.
113. Ajax-Форми.
114. Параметри об'єкту AjaxOptions.
115. AJAX-посилання.
116. Формат JSON.
117. AJAX-запити за допомогою jQuery.
118. Робота з jQuery UI.
119. Авторизація та аутентифікація в MVC.
120. Аутентифікація Windows.
121. Аутентифікація форм.
122. Авторизація в MVC і SimpleMembershipProvider.
123. Налаштування використання SimpleMembershipProvider.
124. Використання універсальних провайдерів.
130. Маршрутизація в Web API.
131. Мобільні додатки на ASP.NET MVC.
132. Метатег Viewport.
133. CSS в мобільних додатках.
134. Створення мобільного додатку на ASP.NET MVC.
135. Режими відображення DisplayMode.
136. Бандли і мініфікація.
137. Створення бандлів.
138. Мультимовний сайт і локалізація.
139. Ресурси. Логіка мультимовного сайту.
140. Фільтр локалізації

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Освітній процес з дисципліни «Веб технології» забезпечений необхідною навчально-методичною літературою. Підготовлено курс у модульному середовищі ХНУ.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Яшина О.М. Веб технології. Методичні вказівки до виконання курсового проекту студентами спеціальності «Інженерія програмного забезпечення». Хмельницький, ХНУ, 2023. – 55 с.
2. WWW Consortium (W3C) Official Page. (<http://www.w3c.org/>).

3. Бернерс-Лі Заснування павутини = Weaving the web. The original design and ultimate destiny of the world wide web : З чого починалася і до чого прийде Всесвітня мережа / Тім Бернерс-Лі разом з Марком Фічетті; пер. з англ. А. Іщенко. – Київ : Києво-Могилянська академія, 2018. – 208 с.
 4. Цеслів О.В. WEB-програмування : навч. посібник / О.В. Цеслів ; М-во освіти і науки, молоді та спорту України, Нац. техн. ун-т України “Київ. політехн. ін-т”. – Київ : НТУУ “КПІ”, 2020. – 149 с.
 5. ASP.NET Core documentation. URL: <https://learn.microsoft.com/aspnet/core/security/authorization/roles?view=aspnetcore-9.0>
 6. What is front-end development. URL: <https://www.freecodecamp.org/news/front-end-developer-what-is-front-enddevelopment-explained-in-plain-english/>.
 7. Front-end development technologies concepts. URL: <https://www.altexsoft.com/blog/front-end-development-technologies-concepts/>.
 8. Beginners guide for back-end development. URL: <https://www.upwork.com/resources/beginners-guide-back-end-development>.
 9. Back-end architecture. URL: <https://www.codecademy.com/article/backend-architecture>
 10. Development tools for web developers. URL: <https://www.geeksforgeeks.org/12-backend-development-tools-for-webdevelopers/>
 11. Який back-end потрібний вашому вебсайту? URL: <https://goldwebsolutions.com/uk/blog/yakij-backend-potribnij-vashomu-vebsajtu/>.
- Допоміжна**
12. Олівер Дік. Популярні Web-браузери. Енциклопедія користувача. – К.:DiaSoft, 2018. – 280 с.
 13. Боуен Річ. Apache. Настільна книга адміністратора. – К.: DiaSoft, 2019. - 350 с.
 14. Матвієнко, О.В. Internet-технології: проектування Webсторінки: Навчальний посібник для студентів вузів / О. В Матвієнко, І. Л Бородкіна. - 2-е вид., перероб. и доп. . - К.: ЦНЛ, 2019. - 154 с.
 15. Куленко М.Я. Основи графічного дизайну : підручник для студентів вищих навч. закладів / Михайло Куленко; МОНУ; Київський нац. ун-т будівництва і архітектури. – 2-ге вид., виправл. та доп. – Київ : Кондор, 2020. – 492с.
 16. GitHub Learning Pathways. URL: <https://resources.github.com/learn/pathways/>.

11. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua>.
2. Електронна бібліотека університету. Доступ до ресурсу: <http://library.khmnu.edu.ua>
3. Репозитарій ХНУ:<https://elar.khmnu.edu.ua/home>



COURSE PROGRAM

Web Technologies

Field of study: 12 - Information Technologies
Major: 121 – Software Engineering
Level of Higher Education: First Level (Bachelor)
Educational program: Software Engineering
Discipline status: Compulsory
Faculty: Information Technologies
Department: Software Engineering

Study mode	Year	Semester	Total Credits	Number of hours						Course project	Coursework	Semester control form	
				ECTS credits	Classwork hours				Seminar classes			Independent work, including individual	pass/ fail test
			Total		Lectures	Laboratory works	Practical classes						
Full-time (Daytime)	3	5	6	68	34	34			112	+		+	
Total			6	68	34	34			112	1		1	

The course program is based on the Higher Education Standard, the 2023 Bachelor's degree educational program, and the curriculum.

Program's author O. Yashyna

Approved at the staff meeting of the Software Engineering Department

Minutes from 31.08.2023 No. 1

Head of the Software Engineering Department L. Bedratyuk

The course program is approved by the Academic Board of the Faculty of Information technologies

Head of the Academic Board O.S. Savenko

WEB TECHNOLOGIES

Type of discipline	Mandatory
Level of higher education	First (Bachelor's)
Language of instruction	Ukrainian, English
Semester	Fifth.
Forms of education	Full-time (Daytime)

The discipline must provide **learning outcomes in** accordance with the Higher Education Standard and the curriculum:

competencies: ability to apply knowledge in practical situations, ability to adhere to specifications, standards, rules, and recommendations in the professional field during the implementation of lifecycle processes, ability to reasonably choose and master the toolkit for software development and maintenance.

programmatic learning outcomes to analyse, purposefully search for, and select the necessary information, reference resources, and knowledge for solving professional tasks, considering modern scientific and technical achievements; to select and utilise a software development methodology appropriate for the task; to understand and apply in practice the fundamental concepts, paradigms, and basic principles of functioning linguistic, instrumental, and computational tools of software engineering; To make informed decisions when choosing programming languages and development technologies to address the tasks of creating and maintaining software.

Content of the discipline (sections). Basic concepts of web development. Addressing on the Internet. GIT version control systems. JavaScript technology. Working with JSON, XML files. Using CMS. The concept of ASP.NET MVC technology. Working with controllers, models, views. Routing. Filters and working with them. Model binding. jQuery and AJAX. Authorisation and authentication in MVC.

Planned educational activities: lectures - 34 hours, laboratory classes - 34 hours, independent work - 112 hours, total - 180 hours.

Teaching methods: lectures (using problem-based learning and visualisation methods), laboratory classes (using computer methods and teaching aids), independent work, course design.

Forms of assessment of learning outcomes: defence of laboratory work, intermediate test controls, defence of a course project.

The form of semester control is an exam.

Educational resources:

1. WWW Consortium (W3C) Official Page. URL: <http://www.w3c.org/>.
2. ASP.NET MVC Pattern. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>.
3. Berners-Lee The founding of the web = Weaving the web. The original design and ultimate destiny of the world wide web: How the World Wide Web Began and Where It Will Go / Tim Berners-Lee with Mark Fischetti; translated from English by A. Ishchenko - Kyiv: Kyiv-Mohyla Academy, 2020. 208 p.
4. Tseslov O.V. WEB-programming: a textbook / O.V. Tseslov: Ministry of Education and Science, Youth and Sports of Ukraine, National Technical University of Ukraine "Kyiv Polytechnic Institute." - Kyiv: NTUU "KPI", 2020. 149 p.
5. Web technologies. Methodical instructions for the implementation of the course project by students of the speciality "Software Engineering". Khmelnytskyi, KhNU, 2023. - 55 p.
6. Rajkumar Buyya, James Broberg, Andrzej Goscinski Cloud computing Principles and Paradigms. John Wiley & Sons, Inc., New Jersey, Canada, 2021. - 637 p.
7. MOODLE Learning Platform. Access to the resource <https://msn.khmnmu.edu.ua>.
8. University Electronic Library. Access to the resource: <http://library.khmnmu.edu.ua>.

Lecturer: Candidate of Technical Sciences, Associate Professor of the Department of IPZ Yashyna O.M.

3. EXPLANATORY NOTE

The discipline "Web Technologies" is one of the professional disciplines and takes a leading place in the training of specialists of the bachelor's degree in the specialty 121 "Software Engineering" under the educational and professional programme "Software Engineering".

Objectives of the discipline: students' mastering of basic knowledge of modern web technologies and basics of web programming, including web services technologies, ASP.NET MVC, mastering of theoretical and practical knowledge on the transformation, transmission and use of information in the global space, revealing the importance of information processes in the formation of a modern system and information scheme of the informatised world.

Subject of discipline: *The subject of the* discipline is the principles of modern programming for the Internet space, as well as methods of their use in the development of web resources for various purposes.

Objectives of the discipline:

To provide students with knowledge and practical skills in developing and implementing web projects using modern programming languages and tools.

In accordance with the Standard of Higher Education in the specified speciality and educational programme, the discipline must provide:

competence:

GC2. Ability to apply knowledge in practical situations.

PC5. Ability to adhere to specifications, standards, rules, and recommendations in the professional field during the implementation of lifecycle processes.

PC13. Ability to reasonably choose and master the toolkit for software development and maintenance.

programme learning outcomes:

PLO1. To analyse, purposefully search for, and select the necessary information, reference resources, and knowledge for solving professional tasks, considering modern scientific and technical achievements.

PLO 6. To select and utilise a software development methodology appropriate for the task.

PLO 7. To understand and apply in practice the fundamental concepts, paradigms, and basic principles of functioning linguistic, instrumental, and computational tools of software engineering.

PLO 15. To make informed decisions when choosing programming languages and development technologies to address the tasks of creating and maintaining software.

4. STRUCTURE OF CREDITS OF THE DISCIPLINE

Topic title	The number of hours allocated to:		
	Lectures	Lab work	Ind. work (including CP)
Topic 1: Methodologies and technologies of web development.	6	8	24
Topic 2. JavaScript technology.	8	8	24
Topic 3: ASP.NET MVC technologies.	20	16	40
Together	34	34	112

5. PROGRAMME OF THE DISCIPLINE

5.1. Content of the lecture course

№ lectures	List of lecture topics and their annotations	Number of hours
1	2	3
1	<p>Topic 1: Methodologies and technologies of web development. Lecture 1: Basic concepts of web development. Functions and tasks of the developer. Web development methodologies. The concept of web applications and components for working with them. WWW service: components, software. Modern web browsers and principles of their work. Overview of CMS systems, frameworks, programming languages, web servers and server-side application programming technologies. Classical and agile web development methodologies: types, principles, rules. Information and reference resources for solving web development problems. [2, 5,7,8]</p>	2
2	<p>Lecture 2: Addressing in the Internet. Data transfer protocols. HTTP/HTTPS protocol. HTTP request methods. Client-server architecture. Transfer of user documents. References: [2, 5,7,8].</p>	2
3	<p>Lecture 3: GIT version control systems. Basic concepts of version control systems. Installing and configuring GIT. Automation of workflows using GitHub Action. [8,9, 16]</p>	2
4	<p>Topic 2. JavaScript technology. Lecture 4. JavaScript basics. Code structure, data types and basic operators. Connecting JavaScript to a web page in different ways. Variables, data types and type conversions. Working with conditional statements JavaScript basics: objects, arrays, loops and functions. methods of objects and arrays. Map and Set collections, WeakMap, WeakSet. Work of loops and data processing for writing business logic of programs. Types of functions and possibilities of code reuse [2, 5, 7, 8].</p>	2
5	<p>Lecture 5. JavaScript: ES5 and ES6 classes. Prototype inheritance. Constructor functions. ES6 classes. OOP in JavaScript: differences from other languages. The 'this' keyword and call context. The main differences between JavaScript OOP and OOP in other languages. Splitting the code into independent modules. [2, 5,7,8]</p>	2
6	<p>Lecture 6. Working with JSON and XML files. Working with different formats in the JavaScript programming language. Queries, event handling and user input. Setting up and sending AJAX requests. Creating DOM elements and adding them to the page. Properly requesting and retrieving data from the server. Handling user events Authorisation and authentication in browsers. Using regular expressions to check the correctness of user input. Different types of authorisations. Cookie and session. Methods of storing data in the browser. [2, 5,7,8]</p>	2
7	<p>Lecture 7. Using CMS. The history of the origin. CMS WordPress, Joomla and others. Differences between CMS and created sites on developed platforms. [3,5,6,7,8].</p>	2
8	<p>Topic 3: ASP.NET MVC technologies. Lecture 8. ASP.NET MVC technologies. The concept of ASP.NET MVC technology. Controller, Model, Model. View. Processing requests by MVC application. Approach to developing MVC applications. Features of ASP.NET MVC. Creating the first application. References: [2,3,5,6,7,8].</p>	2

9	Lecture 9. Working with controllers. Basics of working with controllers. Methods of actions and their parameters. Results of actions. Forwarding and sending status and error codes. Sending files in ASP.NET MVC 5. The context of the request HttpContext. Cookies. Sessions. Asynchronous methods. References: [3,5,6,7,8].	2
10	Lecture 10. Working with representations. Basics of working with views. Strictly typed views. Master pages. Partial views. HTML helpers. Working with forms. Strictly typed helpers. References: [3,5,6,7,8]	2
11	Lecture 11. Working with models. Creating models. Connecting to the database Template helpers. Adding and deleting a model. Formation templates. References: [3,5,6,7,8].	2
12	Lecture 12. Working with models. Models with a complex structure and working with them. Models with many-to-many communication. Transfer arrays and complex data to the controller. References: [3,5,6,7,8].	2
13	Lecture 13. Working with models. Database migration. Creating a pagination. Overriding the formation templates. Overriding editing and image templates. Data filtering. Domain Model and View Model. Managing hierarchical data References: [3,5,6,7,8].	2
14	Lecture 14. Routing. Defining routes. Working with routes. Creating restrictions for routes. Generating outgoing URL addresses. Areas. Create your own route handler. Routing attributes. Routing and nested resources References: [3,5,6,7,8].	2
15	Lecture 15. Metadata and model validation. Data annotations for displaying properties. Basics of validation. Validation attributes. Validation of the model in the controller. Displaying validation errors. Creating your own validation logic. References: [3,5,6,7,8].	2
16	Lecture 16. Filters. Filters in ASP.NET MVC. Authentication filters. Authorisation filters. Model validation in the controller. Displaying validation errors. Creating your own validation logicLit.	2
17	Lecture 17. Authorisation and authentication in MVC. Introduction to authorisation and authentication. Windows authentication. Authentication of forms. Authorisation in MVC. Setting up the use of SimpleMembershipProvider. Using universal providers. Web Site Administration Tool. Overriding the role provider. Profile provider and its overrides. OAuth - authentication. References: [3,5,6,7,8].	2
	Total for the semester	34

5.2. CONTENT OF LABORATORY CLASSES

№ s/n	Topic of the laboratory session	Number of hours
<i>1</i>	<i>2</i>	<i>3</i>
1	Laboratory work №1. Creating a simple application in ASP.NET Core MVC. References: [4, 5].	4
2	Laboratory work №2. Adding a controller, views, and models to a project. Using Create, Read, Update and Delete (CRUD). Reference: [5, 11].	4
3	Laboratory work №3. Choice of user interface. References: [5,6,7].	4
4	Laboratory work №4. Working with the database in applications. References: [5,6,7, 8, 9, 11].	4
5	Laboratory work № 5. Adding search and fields to an ASP.NET Core MVC application. References: [5,6,7].	4
6	Laboratory work № 6. Role-based authorization. References: [5,6,7].	4
7	Laboratory work № 7. Adding validation capabilities to an ASP.NET MVC application. References: [5,6,7].	4
8	Laboratory work № 8. Localization and multilingualism. References: [5,6,7].	4
9	The final lesson.	2
	Total for the semester	34*

Notes. * in the numerator - 34 hours, in the denominator - 32 hours

5.3 CONTENT OF INDEPENDENT WORK

The independent work of *full-time* students consists of systematic study of the programme material, preparation for the performance and defence of laboratory work, testing on theoretical material, completion of individual tasks, including a course project, etc. The course project is carried out in accordance with the methodological recommendations posted in the modular environment, as well as in a printed version at the Department of IPE.

Number of the week	Content of independent work	Number of hours
1	<i>2</i>	<i>3</i>
1	Study of lecture material, preparation for laboratory work No. 1. Determining the topic of the course project.	4
2	Preparation for the defence of laboratory work No. 1 and for laboratory work No. 2. Determination of tasks for the CP.	4
3	Study the lecture material. Preparation for laboratory work #2. Study of reference literature. Analyse the subject area in accordance with the subject matter of the CP.	4
4	Study of lecture material. Preparation for the defence of laboratory work No. 2. Analyse existing software solutions in accordance with the subject of the course.	4

5	Study the lecture material. Preparation for laboratory work No. 3. Study of reference literature. Development of TK for the project.	4
6	Study of lecture material. Preparation for the defence of laboratory work No. 3. Study of reference literature. Designing the structure of the programme in accordance with the subject matter of the CP.	6
7	Study the lecture material. Preparation for laboratory work No. 4. Study of reference literature. Designing the architecture of the software system in accordance with the subject of the control panel.	6
8	Study of lecture material and reference books. Preparation for TC1. Designing a database in the programme in accordance with the subject of the CP.	6
9	Study of lecture material. Preparation for the defence of laboratory work № 4. Choosing the means and technologies for implementing the programme in accordance with the subject matter of the CP.	6
10	Study the lecture material. Preparation for laboratory work No. 5. Study of reference literature. Implementation of the programme in accordance with the subject	6
11	Study the lecture material. Preparation for laboratory work No. 6. Study of reference literature. Implementation of the programme in accordance with the subject matter of the CP.	6
12	Study of lecture material. Preparation for the defence of laboratory work No. 6. Study of reference literature. Implementation of the programme in accordance with the subject matter of the CP.	6
13	Study the lecture material. Preparation for laboratory work No. 7. Study of reference literature. Writing a user manual.	6
14	Study of lecture material. Preparation for the defence of laboratory work № 7. Study of reference literature. Testing of the KP programme.	6
15	Study the lecture material. Performing laboratory work №8. Study of reference literature. Prepare for the test and send the assignment to the teacher.	6
16	Study the lecture material. Preparation for the defence of laboratory work №8. Study of reference literature. Preparation of the explanatory note of the CP and checking for anti-plagiarism.	6
17	Study of lecture material. Preparation for TC2. Preparation for the defence of the CP.	6
Together		112

6. TEACHING METHODS

The learning process in the discipline is based on the use of traditional and modern methods. In particular, lectures are conducted mainly by verbal methods, and laboratory classes are conducted using information technology, master classes, workshops and are aimed at acquiring practical skills in creating a software product using modern methods.

7. FORMS AND METHODS OF ASSESSMENT OF LEARNING OUTCOMES

Current control is carried out during lectures and laboratory classes, as well as on the days of control measures established by the work plan of the discipline. Semester control is carried out in the form of a test. The results of the current control are taken into account when determining the final grade.

The following methods of current control are used:

- oral questioning of students before admission to the laboratory class;
- admission to the defence of laboratory work;
- protection of laboratory work;
- testing and verification of the operability of software, hardware and other software created in the course of the laboratory work;

The final score is based on the weighting factors.

A student who is not admitted to the defence or has not defended laboratory work is considered to be failing.

The process of assessing a student's readiness can be divided into stages:

The first stage of assessment is aimed at determining the knowledge of the information minimum. If a student has firmly mastered the amount of formal knowledge specified in the curriculum, it means that he or she is able to use it in solving various issues in the design of leather goods and is able to expand it.

Before studying a discipline, as a rule, an input control of knowledge of the disciplines that precede and provide it is carried out. In this case, it is necessary to establish the levels and criteria for the formation of knowledge about the content of the educational elements. These levels are:

Introductory-oriented (IO) - a person has an approximate understanding of the concepts studied, is able to: reproduce the wording of the definitions of the concept of mobile platforms; navigate the methods of software product development; know the theoretical foundations of creating applications for mobile platforms.

Conceptual-analytical (CA) - a person has a clear understanding of the learning object, is able to make semantic distinction, explain the choice of constructive solutions for creating a software product, methods of their design, their justification based on the analysis of existing ones. Can clearly define the structure of programmes when designing them depending on their use, i.e. able to transfer previously acquired knowledge to typical situations.

Productive-synthetic (PS) - a person has a deep understanding of the learning object, is able to carry out synthesis, generate new ideas and concepts, transfer previously acquired knowledge to atypical, non-standard situations. That is, at this level, a student should be able to design various types of software products using the most common methods, make suggestions for improving the methods of designing and developing applications for mobile platforms, and create competitive software applications in the global market based on theoretical knowledge.

Various means of control are used to assess students' knowledge, in particular:

-Admission to the laboratory work is carried out at the beginning of the laboratory work by oral questioning of the student;

-The quality of the laboratory work is assessed by checking the completed laboratory work in the working file and in the workbook;

-The acquisition of theoretical knowledge and practical skills is tested through the defence of each laboratory work;

-Mastery of theoretical material is checked by test control.

The grade assigned for a laboratory class consists of the following elements: oral questioning of students before admission to the laboratory work; knowledge of theoretical material on the topic; quality of the results of the laboratory work in the notebook and in the working file; student's ability to justify decisions and draw conclusions; timely defence of the laboratory work.

The deadline for defending a laboratory work is considered timely if the student defends it at the next class after the work is completed.

A student must make up a laboratory lesson missed for a valid reason within the time limit set by the teacher.

Each type of work is graded on a four-point scale. The final grade in the discipline is assigned as a weighted average of all types of work.

Table 1. Structuring the discipline by types of work and assessing the learning outcomes of full-time students in a semester by weighting coefficients

Audit work			Independent, individual work		Semester control
<i>V semester</i>					
Laboratory work no:			Test control:		Exam
1-8			TC 1-2		1
WK*: 0,4			0,2		0,4
<i>V semester (course project)</i>					
<i>Control points №</i>					Defence CP
1 section	2 section	3 section	4 section	project	1
WK: 0,1	WK: 0,1	WK: 0,1	WK: 0,1	WK: 0,3	WK: 0,3

Abbreviations: TC - test control; WK - weighting factor.

Assessment of student work according to the traditional system

The grade of "*excellent*" is given to a student who is fluent in the material, clearly, competently, logically, fully and consistently covers the issue, demonstrates knowledge not only of the lecture material, but also of the results of independent study of the literature, the ability to present their own reasoning and draw conclusions.

The highest grade is given to a student who has demonstrated comprehensive and deep knowledge of the curriculum material, understands the concepts, principles and methods of analysing software requirements, as well as: who has mastered the main and is familiar with additional literature, has mastered the basic concepts of the discipline; has given complete and accurate answers to all questions with explanations and conclusions; has shown creativity in understanding, presenting and using the educational material.

A grade of "*good*" is given to a student who has a good command of the academic material, presents it competently and logically in oral or written form, making no more than one or two shortcomings or minor mistakes, limits himself to the presentation of the main theoretical provisions, does not know the material of additional literature, although he generally has solid knowledge.

"Satisfactory is awarded to a student who has demonstrated knowledge of the basic material to the extent necessary for further study (basic concepts, principles, patterns), and who is familiar with the basic literature; feels uncertain when answering questions, makes minor mistakes; and has the knowledge to correct them under the guidance of a teacher when making mistakes in answers.

A grade of "*unsatisfactory*" is assigned to a student who demonstrates incomplete knowledge of the course or its individual topic, makes significant mistakes in the presentation of the course material, and is not familiar with the lecture material and the material of the main and additional literature.

An "*unsatisfactory*" grade is assigned to a student who has gaps in knowledge of the basic material, and who: has made fundamental mistakes, does not understand the basic conceptual provisions of the discipline; does not give satisfactory answers to questions, does not cope with solving problems; cannot continue his or her studies without additional classes in the discipline.

If a student receives a negative grade, he or she must retake it in accordance with the established procedure, but always before the next control. If a student fails to complete an individual plan for a

discipline within the scheduled timeframe without valid reasons, then, if the student gives a positive answer, he or she will be given a "satisfactory" grade.

The final semester grade according to the national and ECTS scales is set in an automated mode after all grades are entered into the electronic journal. The ratio of the national grading scale and the ECTS grading scale is shown in Table 2.

A credit is awarded if the weighted average score received by a student in a discipline is within the range of 3.00 to 5.00 points. In this case, the national scale is "passed", and the ECTS scale is the letter grade corresponding to the number of points gained by the student.

Table 2. Correlation between the national grading scale and the ECTS grading scale

ECTS assessment	Points	Domestic system	
A	4,75-5,00	5	EXCELLENT - deep and complete mastery of the educational material and identification of relevant skills
B	4,25-4,74	4	GOOD - complete knowledge of the study material with a few minor errors
C	3,75-4,24	4	GOOD - generally correct answer with two or three significant errors
D	3,25-3,74	3	SATISFACTORY - incomplete mastery of the programme material, but sufficient for practical work in the profession
E	3,00-3,24	3	Satisfactory - incomplete mastery of the programme material that meets the minimum assessment criteria
FX	2,00-2,99	2	UNSATISFACTORY - unsystematic knowledge and inability to continue studying without additional knowledge of the discipline
F	0,00-1,99	2	UNSATISFACTORY - serious further work and re-study of the discipline is required

8. QUESTIONS FOR SELF-MONITORING OF LEARNING OUTCOMES

1. What is a web application?
2. What is a browser?
3. Describe the cycle of processing a request to a web application from a client.
4. Why do you need web application development technologies (such as ASP.NET, PHP, Ruby On Rails, etc.)?
5. How does the HTTP protocol work and what is it for?
6. What are HTTP message headers and what are they for?
7. What is the body of an HTTP message?
8. How are headers separated from the message body in an HTTP message?
9. What is the HTTP request method?
10. What is the HTTP response status code?
11. Give examples of HTTP headers for an HTTP request and an HTTP response.
12. How does the secure HTTPS protocol work?
13. What is a web server?
14. What interfaces can a web server and a web application use to communicate?
15. What is the difference between CGI and ISAPI?
16. What is shared hosting?
17. What is an application pool?
18. Name the most popular implementations of web servers.

19. What kind of web server do ASP.NET applications run on?
20. Name the stages of MVC application development.
21. Describe the request lifecycle for an MVC application.
22. Microsoft ASP.NET MVC is a ?
23. What is a Controller?
24. What is the Model?
25. What is a Submission?
26. The algorithm of the MVC application.
27. What tasks is the ASP.NET MVC platform designed to solve?
28. The extension methods are LINQ.
29. The from construct and range variables are LINQ.
30. The join construction is LINQ.
31. The orderby construction is LINQ.
32. The select construction and projection are LINQ.
33. The let construct is LINQ.
34. The group construction is LINQ.
35. The in and continuation constructs are LINQ.
36. The basics of controllers.
37. Defining the IController interface.
38. Methods of action.
39. Receiving input data. Transferring parameters.
40. Classes derived from ActionResult. 60. Passing data from the controller to the view.
41. Redirection. Objects RedirectResult, RedirectPermanent.
42. The RedirectToAction method.
43. The ContentResult.
44. The context of the request.
45. Cookies.
46. Introduction to the view.
47. Representation using Razor syntax.
48. Paths to view files.
49. Types of presentation templates.
50. Strictly typed representations.
51. Razor view engine.
52. Master pages.
53. The _ViewStart.
54. Partial representations.
55. Helper Html.Partial.
56. HTML headers.
57. Strictly typed helpers. Lambda expressions.
58. Models and databases.
59. Entity Framework.
60. Connecting to the database.
61. Template helpers.
62. Edit the model.
63. Form templates.
64. Models with a complex structure.
65. Working with complex models.
66. Data transfer to the controller.
67. Routing.
68. Defining routes.
69. Working with routes.
70. Create restrictions for routes.
71. Generate outgoing URLs.

72. Areas.
73. Create your own route handler.
74. Metadata.
75. Model validation.
76. Data annotations for displaying properties.
77. Validation attributes.
78. Validation of the model in the controller.
79. Display of validation errors.
80. Create your own validation logic.
81. Filters.
82. Authorisation filters.
83. Exception filters.
84. Filters for actions and results.
85. Global filters.
86. Built-in filters.
87. Linking the model.
88. Create a model binder.
89. jQuery and AJAX.
90. 12. Using JavaScript/jQuery.
91. Ajax Forms.
92. Parameters of the AjaxOptions object.
93. AJAX link.
94. JSON format.
95. AJAX requests using jQuery.
96. Working with jQuery UI.
97. Authorisation and authentication in MVC 4.
98. Windows authentication.
99. Authentication of forms.
100. Authorisation in MVC and SimpleMembershipProvider.
101. Setting up the use of SimpleMembershipProvider.
102. Use of universal providers.
103. Routing in the Web API.
104. Creating and testing a Web API application.
105. Create a view for the Web API.
106. Mobile applications in ASP.NET MVC 4.
107. The Viewport meta tag.
108. CSS in mobile applications.
109. Creating a mobile application in ASP.NET MVC.
110. Display modes DisplayMode.
111. Bundles and minification.
112. Creating gangs.
113. Multilingual website and localisation.
114. Resources. The logic of a multilingual website.
115. Localisation filter

9. METHODOLOGICAL SUPPORT

The educational process in the discipline "WEB TECHNOLOGIES" is provided with the necessary educational and methodological literature. The course was prepared in the modular environment of KhNU.

10. RECOMMENDED READING

Main

1. Web technologies. Methodical instructions for the implementation of the course project by students of the speciality "Software Engineering". Khmelnytskyi, KhNU, 2023. 55 p.
2. WWW Consortium (W3C) Official Page. (<http://www.w3c.org/>).
3. Berners-Lee The founding of the web = Weaving the web. The original design and ultimate destiny of the world wide web: How the World Wide Web Began and Where It Will Go / Tim Berners-Lee with Mark Fischetti; translated from English by A. Ishchenko - Kyiv: Kyiv-Mohyla Academy, 2018. 208 p.
4. Web programming: a textbook / O.V. Tseslov; Ministry of Education, Youth and Sports of Ukraine, National Technical University of Ukraine "Kyiv Polytechnic Institute." - Kyiv: NTUU "KPI", 2020. 149 p.
5. ASP.NET Core documentation. URL: <https://learn.microsoft.com/aspnet/core/security/authorization/roles?view=aspnetcore-9.0>
6. What is front-end development. URL: <https://www.freecodecamp.org/news/front-end-developer-what-is-front-enddevelopment-explained-in-plain-english/>.
7. Front-end development technologies concepts. URL: <https://www.altexsoft.com/blog/front-end-development-technologies-concepts/>.
8. Beginners guide for back-end development. URL: <https://www.upwork.com/resources/beginners-guide-back-end-development>.
9. Back-end architecture. URL: <https://www.codecademy.com/article/backend-architecture>.
10. Development tools for web developers. URL: <https://www.geeksforgeeks.org/12-backend-development-tools-for-webdevelopers/>.
11. What back-end does your website need? URL: <https://goldwebsolutions.com/uk/blog/yakij-backend-potribnij-vashomu-vebsajtu/>.

Auxiliary

12. Oliver Dick. Popular Web browsers. User's encyclopedia. - K.: DiaSoft, 2018. - 280 p.
13. Bowen Rich. Apache. Administrator's Handbook - K.: DiaSoft, 2019. - 350 p.
14. Matvienko, O.V. Internet-technologies: designing a web page: A textbook for university students / O.V. Matvienko, I.L. Borodkina. - 2nd edition, revised and supplemented. - K.: TSNL, 2019. - 154 p.
15. Kulenko M.Y. Fundamentals of graphic design: a textbook for students of higher educational institutions / Mykhailo Kulenko; MESU; Kyiv National University of Construction and Architecture. 2nd edition, revised and supplemented - Kyiv: Condor, 2020. 492 p.
16. GitHub Learning Pathways. URL: <https://resources.github.com/learn/pathways/>.

11. INFORMATION RESOURCES

1. MOODLE Learning Platform. Access to the resource <https://msn.khmnu.edu.ua>.
2. University Electronic Library. Access to the resource: <http://library.khmnu.edu.ua>.
3. University Repository. Access to the resource <https://elar.khmnu.edu.ua/home>.