



Силабус дисципліни «Прогресивні web-застосунки (PWA)»

Викладач: Бурлаченко Іван Сергійович
старший викладач кафедри комп'ютерної інженерії
факультету комп'ютерних наук ЧНУ імені Петра Могили.

Очікувані результати навчання

В результаті вивчення дисципліни студент

має знати:

- об'єктно-орієнтовану модель мов JavaScript та TypeScript і особливості проектування програмних компонентів;
- основні архітектурні особливості програмного забезпечення, розробленого на React та Angular;
- як організувати декомпозицію коду проекту на складові компоненти для зручності подальшої підтримки та повторного використання;
- як вирішувати системні завдання під час web-розробки на рівні, достатньому для промислового застосування;

має вміти:

- впроваджувати програмні залежності у конфігурацію React та Angular проєктів;
- проектувати моделі сервісів даних у React та Angular застосунках і налаштовувати web socket з'єднання;
- кешувати шаблони, відображати динамічні дані, створювати призначені для користувача функції і обробляти помилки під час виконання;
- використовувати і налаштовувати сеанси для збереження даних між запитами;
- налаштовувати React та Angular застосунки для підвищення продуктивності і безпеки програмних продуктів;
- впровадити аутентифікацію і авторизацію, та запобігати SQL

Обсяг: буде визначено після здійснення процедури вибору студентами вибіркового дисциплін.

Мета: навчитися використовувати системний підхід до проектування архітектури великих програмних продуктів, працювати в командах front-end розробників.

Оригінальність навчальної дисципліни

Матеріали дисципліни представляють сукупний досвід інженерії програмного забезпечення багатьох tech-лідерів проєктів та solution-архітекторів міжнародних ІТ компаній.

Зміст дисципліни

Тема 1. Основи ES6. Налаштування оточення.Короткий огляд особливостей ES6. Створення React app.

Тема 2. React. Вступ до ReactJS. Бібліотека React.Що таке реактивний елемент? Знайомство з JSX. Компоненти.

Тема 3. React. Віртуальний DOM.Stateful-компоненти, зберігання стану.State, props.

Тема 4. React. Віртуальний DOM. Як React відстежує зміни (reconciliation).

Тема 5. React. Життєвий цикл react-компонентів.

Тема 6. React. Огляд react-компонентів. "Розумні" компоненти. Синтетичні події.

Тема 7. React. Керовані і некеровані компоненти. Ієрархія компонентів. Контекст.

Тема 8. React. React Router. Завдання React Router. Основні версії, їх відмінності.

Тема 9. React. Налаштування роутера. Перенаправлення, лінива завантаження. Управління роутером.

Тема 10. React. Redux. Чим Redux відрізняється від FLUX. Потік даних. Middleware.

Тема 11. React. Архітектура React + Redux застосунків. Бібліотека react-redux.

Тема 12. React. Компоненти-контейнери і презентаційні компоненти. Стили.

Тема 13. React. Селектори. Імутабельність. Структура проєкту.

Тема 14. React. Вступ в хуки (Hooks). Використання хука стану. Використання

ін'єкціям, CSRF, XSS, перехопленню кліків.

Тема 15. React. Правила хуків. Створення хуків, що призначені для користувача. API хуків.

Тема 16. React. Огляд конкурентного режиму. Затримка при отриманні даних. Concurrent UI Patterns.

Тема 17. Angular. Вступ до Angular. Знайомство з фреймворком. Основні переваги.

Тема 18. TypeScript. Унікальні можливості. Знайомство з TypeScript. Основні відмінності. Відмінності від попередніх версій.

Тема 19. Angular. Знайомство з розробкою на Angular. Структура фреймворку. Опис і функціонал основних модулів.

Тема 20. Angular. Особливості створення проєкту. Angular guidelines + TSLint introduction.

Тема 21. Angular. Основи роботи з CLI. Ініціалізація проєкту. Види шаблонів.

Тема 22. Angular. Створення компонентів. Створення білдів. Основи роботи з DOM в Angular.

Тема 23. Angular. Компоненти. Директиви. Pipes.

Тема 24. Angular. Шаблонізація в Angular. Основи створення Angular Web Components - Angular Elements.

Тема 25. Angular. Основи роботи з даними в Angular. Декоратори. Сервіси. Прив'язка даних.

Тема 26. Angular. Збереження даних. Взаємодія між компонентами. Робота з формами в Angular.

Тема 27. Angular. Основні життєві цикли додатки на Angular. Методи OnInit і OnDestroy. Конструктор в компонентах застосунків.

Тема 28. Angular. Спадкування, інкапсуляція властивостей і методів компонента.

Тема 29. Angular. Роутінг PWA в Angular. Створення модуля роутінга для застосунку. Директива Router Outlet.

Тема 30. Angular. Директиви для роботи з роутером. Створення URL з параметрами і їх обробка. Основи роботи з HTTP.

Тема 31. Angular. Розбір модуля HttpClient API. Інтерфейси для

хука ефекту.

Тема 32. Angular. Створення запитів до сервера з використанням Promises.

Особливості роботи Observables. частин програми.

Пререквізити

Програмування, об'єктно-орієнтоване програмування, алгоритми і структури даних

Постреквізити

інженерія програмного забезпечення, крос-платформне програмування, web-технології, технології розробки мобільних застосунків, паралельне програмування, методи машинного навчання, формальні граматики та компілятори, розпізнавання образів, системи та методи прийняття рішень.

Семестровий контроль: залік/іспит

Оцінювання:

За семестр: 70/60 балів

За залік/іспит: 30/40 балів

Види робіт:

Лабораторні роботи – 60/50 балів (12/10 x 5 б.)

Індивідуальне проєктне завдання – 10 балів.

Технічне забезпечення

Modern browsers

VS Code IDE

Moodle, Google Meet, Zoom, Webex, профілі соціальних мереж

Політика щодо дедлайнів

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.

Політика щодо академічної доброчесності

Передбачає самостійне виконання лабораторних робіт та індивідуального проєктного завдання. Списування під час заліку (в т. ч. із використанням мобільних пристроїв) заборонено. У разі виявлення плагиату або списування роботи не

роботи з усіма методами HTTP запитів.

Критерії оцінювання лабораторних робіт

- вміння самостійно вирішувати поставлені задачі та представляти рішення у вигляді оптимального програмного коду.
- вміння користуватися специфічними інструментальними засобами для створення програмного забезпечення, що задовільняє SOLID принципам.
- вміння лаконічно та логічно формувати відповіді на запитання, пов'язані з виконаними лабораторними роботами.

зараховуються.

Критерії оцінювання індивідуального проєктного завдання

- формулювання актуальності, проблеми, мети і завдань, практичного значення;
- вміння самостійно створювати оптимальний програмний код для вирішення поставленої задачі з використанням паттернів програмування.;
- вміння користуватися специфічними інструментальними засобами для розробки ПЗ з ефективною архітектурою;
- якість доповіді студента (форма доповіді, зміст, доказова база, висновки) ;
- повнота та логічність відповідей на поставлені питання.