Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

И.В. Безходарнов

ВЕБ-РАЗРАБОТКА

Методические указания
для проведения практических занятий

Томск 2025

УДК 004.021
ББК 32.973.4
Б39

Рецензент:
Гриценко Ю.Б., кафедра АОИ ТУСУР, к.т.н., доцент

Автор: И.В. Безходарнов

Безходарнов, Илья Владимирович
Б39 Веб-разработка: метод. указания для проведения практических занятий / И.В. Безходарнов – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2025. – 12 с.

Методические указания предназначены для проведения практических занятий по дисциплине «Веб-программирование». Пособие охватывает три ключевых этапа разработки веб-приложений: создание серверной части с использованием FastAPI, реализация клиентского интерфейса на HTML и JavaScript, а также размещение приложения в облаке с помощью Docker и облачных платформ.

Каждое занятие включает теоретическое обоснование, пошаговое практическое задание и описание ожидаемых результатов. Методичка нацелена на развитие у студентов целостного понимания архитектуры современных веб-сервисов и приобретение навыков, необходимых для профессиональной деятельности в области веб-разработки.

Одобрено на заседании ПИШ, протокол № 9 от 03.09.2024 г.

УДК 004.021
ББК 32.973.4

Оглавление

Введение

Современные веб-приложения представляют собой сложные программные комплексы, включающие в себя клиентскую (фронтэнд) и серверную (бекэнд) части, взаимодействующие по сети. Для специалистов в области компьютерных наук важно не только понимать архитектуру веб-приложений, но и уметь разрабатывать и развертывать такие системы с использованием актуальных технологий.

Цель настоящих методических указаний — сформировать у студентов практические навыки по созданию полноценных веб-приложений, начиная с разработки серверной логики, создания пользовательского интерфейса и заканчивая публикацией приложения в сети. В рамках курса уделяется внимание следующим технологиям: FastAPI — современный фреймворк для построения API на Python, и JavaScript — для взаимодействия с сервером на стороне клиента. Практические занятия направлены на закрепление теоретических знаний, развитие технического мышления и освоение инструментов, востребованных в индустрии.

Методические указания включают три практических занятия:

1. Разработка бекэнд части веб-приложений с использованием FastAPI;
2. Разработка фронтэнд части с использованием HTML и JavaScript;
3. Размещение веб-приложения в облаке и обеспечение его эксплуатационной готовности.

Каждое занятие состоит из краткого теоретического обзора, пошагового описания практического задания и перечня ожидаемых результатов.

Тема №1 Разработка бекэнд части веб-приложений

Цель: научится устанавливать, конфигурировать и запускать веб-проект с использованием фреймворка FAST API на Python.

Задачи:

1. Изучить документацию на FAST API в объеме, достаточном для выполнения задания по теме.
2. Установить и настроить FAST API, запустить его в режиме «по умолчанию».
3. Разработать учебный проект на языке Python в виде Web API, для проверки работоспособности использовать встроенную в фреймворк систему интерактивной документации Swagger.

Теоретическая часть

Бекэнд (backend) — серверная часть веб-приложения, отвечающая за обработку логики, работы с базой данных, управление пользователями и обеспечение безопасности. Одним из современных и популярных фреймворков для разработки API на Python является FastAPI. Он позволяет быстро создавать RESTful API с поддержкой асинхронных запросов, автоматической валидацией данных и документацией OpenAPI.

Основные компоненты FastAPI:

- Декораторы для маршрутов: @app.get(), @app.post() и др.
- Асинхронные функции обработки запросов.
- Встроенная поддержка моделей данных через Pydantic.
- Интеграция с базами данных через ORM (например, SQLAlchemy).

Практическая часть

Создать простой REST API для работы с задачами (TODO list), включающий:

- получение списка задач;
- добавление новой задачи;
- обновление существующей задачи;
- удаление задачи.

Этапы выполнения:

1. Установка и настройка FastAPI и Uvicorn.
2. Определение модели данных с использованием Pydantic.
3. Реализация маршрутов для CRUD-операций.
4. Запуск и тестирование сервиса локально.

5. Автоматическая документация Swagger.

Полученные результаты

После выполнения практического задания вы должны:

- Уметь развернуть минимальный FastAPI-проект.
- Реализовать REST API с базовыми CRUD-операциями.
- Использовать Pydantic для валидации данных.
- Понимать структуру backend-части современного веб-приложения.

Тема №2 Разработка фронтэнд части веб-приложений

Цель: разработать учебное ВЕБ-приложение, взаимодействующее с API через HTTP запросы и пользователем через интерфейс, созданный с помощью HTML/CSS/JavaScript.

Задачи:

1. Подготовить структуру ВЕБ-приложения с использованием HTML/CSS/JavaScript.
2. Создать часть приложения, отвечающую за обмен данными с сервером через API.
3. Создать пользовательский интерфейс, обеспечивающий взаимодействие пользователя с ВЕБ-приложением.

Теоретическая часть

Фронтэнд (frontend) — клиентская часть веб-приложения, реализуемая с использованием HTML, CSS и JavaScript. В рамках дисциплины рассматривается реализация взаимодействия с REST API средствами JavaScript через fetch API. Также важны понятия DOM, событий и валидации пользовательского ввода.

Основные элементы:

- HTML-структура (формы, таблицы, кнопки).
- JavaScript для отправки HTTP-запросов.
- Обработка событий и асинхронные функции.
- Взаимодействие с JSON-ответами API.
- Отображение данных в интерфейсе.

Практическая часть

Создать одностраничное веб-приложение для работы с API, разработанным на предыдущем занятии.

Функциональность:

- Отображение списка задач с сервера.
- Добавление новой задачи через форму.
- Возможность отметить задачу как выполненную.
- Удаление задачи.

Этапы выполнения:

1. Создание HTML-шаблона со списком задач и формой;

2. Реализация загрузки данных с сервера с помощью fetch;
3. Обработка событий форм (submit, click);
4. Динамическое обновление DOM после изменения данных;
5. Обработка ошибок.

Полученные результаты

По итогам занятия вы должны:

- Владеть основами работы с DOM и событиями в JavaScript.
- Уметь отправлять и обрабатывать HTTP-запросы fetch.
- Интегрировать фронтэнд с REST API.
- Понимать принципы одностраничных приложений (SPA).

Тема №3 Размещение веб-приложений, обслуживание и эксплуатация

Цель: ознакомится на практике с технологиями виртуализации с помощью Docker контейнеров, созданием серверной среды для запуска приложений.

Задачи:

1. Запуск приложения, полученного на предыдущих занятиях в Docker контейнерах.
2. Размещение исходного кода на GitHub (или аналогичной платформе).
3. Развертывание приложения на сервере с помощью Docker контейнеров.

Теоретическая часть

После разработки веб-приложения его необходимо разместить на сервере и обеспечить его работоспособность.

Современные способы размещения:

- Облачные платформы (Render, Heroku, Railway).
- VPS/облачные машины (DigitalOcean, AWS).
- Использование Docker для контейнеризации.

Основные задачи эксплуатации:

- Настройка серверной среды (Python, Uvicorn, Nginx).
- Обеспечение безопасности (HTTPS, CORS, защита от SQL-инъекций).
- Мониторинг и логирование ошибок.
- Резервное копирование данных.

Практическая часть

Развернуть созданное веб-приложение (FastAPI + HTML/JS) в облаке.

Этапы выполнения:

1. Подготовка Dockerfile для backend и frontend.
2. Сборка и тестирование контейнеров локально.
3. Загрузка проекта на GitHub.
4. Развертывание с помощью Render или Railway.
5. Проверка работы API и фронтэнда по URL.

Полученные результаты

В результате третьего занятия должны вы должны:

1. Уметь контейнеризовать приложение с помощью Docker.
2. Размещать и настраивать сервер для развертывания приложения.
3. Обеспечить доступность и работоспособность проекта онлайн.
4. Понимать основы DevOps для веб-разработки.

Заключение

Публичный репозитарий с исходным кодом, написанного вами приложения может служить примером вашей работы, которым можно поделиться для представления вашего пусть небольшого, но реального опыта и навыков работы по созданию приложений.

Изучение и практическое освоение всех этапов разработки веб-приложений — от создания серверной логики и клиентского интерфейса до размещения в сети — позволяет студентам сформировать системное представление о современных подходах к веб-программированию.

Данные методические указания направлены на развитие комплексных навыков, включающих проектирование API, взаимодействие с фронтэндом, базовые принципы DevOps и работу с облачными платформами. Выполнение практических заданий обеспечивает понимание ключевых технологических компонентов веб-разработки, а также формирует уверенность в создании и развертывании полноценных веб-сервисов.

Настоящий курс является хорошей основой для дальнейшего изучения более сложных архитектурных подходов (например, микросервисов), работы с фреймворками фронтэнд-разработки, системами аутентификации, а также глубокой интеграции с базами данных и внешними сервисами.

Список рекомендуемой литературы

1. FastAPI : сайт / tiangolo. – URL: https://fastapi.tiangolo.com/. Режим доступа свободный (дата обращения 15.08.2024).

2. JavaScript tutorial : сайт / Refsnes Data. – URL: https://www.w3schools.com/js. Режим доступа свободный (дата обращения 15.08.2024).

3. *Полуэктова, Н. Р.* Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 204 с. https://urait.ru/bcode/567610 (дата обращения 29.08.2024)