Слайд 1: Что такое веб-приложение?

Веб-приложение — это сайт, с которым можно взаимодействовать. Например, вы заказываете еду онлайн или отправляете сообщения в социальной сети. Веб-приложения позволяют пользователям выполнять различные действия, такие как покупка товаров, общение с друзьями или бронирование билетов.

Слайд 2: Примеры веб-приложений

Примеры веб-приложений включают онлайн-магазины и социальные сети. В онлайн-магазине вы можете искать товары, добавлять их в корзину и делать заказы. В социальной сети вы можете публиковать посты, ставить лайки и подписываться на друзей. Эти примеры помогут вам лучше понять, как работают веб-приложения.

Слайд 3: Как работает веб-приложение?

Веб-приложение состоит из двух частей: клиентской и серверной. Клиентская часть — это то, что видит пользователь, а серверная часть — это то, что скрыто от пользователя. Обе части работают вместе, чтобы обеспечить функциональность веб-приложения.

Слайд 4: Клиентская часть

Клиентская часть — это страничка в браузере. Например, когда вы заходите на сайт магазина и видите каталог товаров. Она отвечает за внешний вид и то, как вы взаимодействуете с сайтом. Клиентская часть включает в себя HTML, CSS и JavaScript, которые обеспечивают отображение и интерактивность страницы.

Слайд 5: Серверная часть

Серверная часть — это то, что скрыто от пользователя. Сервер обрабатывает запросы, например, когда вы ищете товар в каталоге или отправляете сообщение в соцсети. Сервер также хранит данные, например, информацию о товарах или постах пользователей. Серверная часть может быть написана на различных языках программирования, включая Python.

Слайд 6: Простое объяснение компонентов веб-приложений

Клиент — это как пользователь, который задает вопросы. Например, "Покажи мне все товары для сада". Клиент отправляет запросы на сервер, чтобы получить нужную информацию или выполнить определенные действия.

Слайд 7: Примеры из проектов

Онлайн-магазин товаров для дома и сада:

Бэкенд для онлайн-магазина товаров для дома и сада с использованием фреймворка Django Rest Framework. Бэкенд работает на основе архитектуры RESTful API, обрабатывает запросы с фронтенда и хранит информацию в базе данных. В рамках разработки онлайн-магазина вы реализуете фичи: регистрации и авторизации, каталог товаров и пагинация, фильтрация товаров по категориям, цене, названию и т.д., добавление в избранное и в корзину, отслеживание статуса заказа. Реализуете роли пользователя и администратора онлайн-магазина. Покроете приложение тестами. Реализуете приложение и в монолитной, и в микросервисной архитектуре. После модуля по нейросетям добавите рекомендательную систему по выдаче товаров пользователю на основе его истории покупок.

Слайд 8: Примеры из проектов (продолжение)

Социальная сеть Chatty:

"Chatty" — это социальная сеть, предназначенная для публикации контента пользователями, а также для взаимодействия с контентом других пользователей (лайки, комментарии, подписки). Вы реализуете функционал регистрации и авторизации пользователей, создание и редактирование профилей, публикацию постов в блоге, просмотр чужих постов, а также возможность обратной связи с разработчиками. Создадите роли пользователей и администраторов. Администраторам предоставляются расширенные возможности по управлению аккаунтами и контентом. Приложение реализуется в двух версиях: с монолитной и микросервисной архитектурой. Функционал покрывается юнит-тестами и интеграционными тестами. После модуля по нейросетям добавите рекомендательную систему по добавлению в друзья.

Слайд 9: Что будут делать студенты?

Онлайн-магазин: создадите систему, где люди смогут искать товары, добавлять их в корзину и делать заказы. Вы научитесь создавать интерфейс для пользователей, обрабатывать их запросы и управлять данными о товарах и заказах.

Слайд 10: Что будут делать студенты? (продолжение)

Социальная сеть: создадите систему, где пользователи смогут делиться постами, лайкать, подписываться на друзей. Вы научитесь создавать интерфейс для пользователей, обрабатывать их запросы и управлять данными о профилях и постах.

Слайд 11: Как студенты создадут эти приложения?

Шаг 1: Начнём с основ Python и научимся писать простые программы. Мы изучим основные концепции Python, такие как переменные, циклы, функции и модули. Это даст вам базовые навыки для создания более сложных приложений.

Слайд 12: Как студенты создадут эти приложения? (продолжение)

Шаг 2: Постепенно перейдём к созданию серверной части, чтобы обрабатывать запросы. Мы изучим, как создавать серверные приложения на Python с использованием фреймворка Django. Это позволит вам обрабатывать запросы пользователей и взаимодействовать с базой данных.

Слайд 13: Как студенты создадут эти приложения? (продолжение)

Шаг 3: Научимся сохранять и доставать информацию из базы данных, например, список товаров или постов. Мы изучим, как работать с базами данных, такими как SQLite или PostgreSQL, и как использовать ORM (Object-Relational Mapping) для упрощения работы с данными.

Слайд 14: Заключение

После этого курса вы сможете создавать веб-приложения: от простого интернет-магазина до социальной сети. Вы получите навыки, которые помогут вам создавать интерактивные и функциональные веб-приложения.