

Лабораторная работа 1 (1 занятие)

Тема: Архитектура веб-приложения.

Теория

Архитектура веб-приложений — это схема взаимодействия между компонентом, базой данных, сервером и системой промежуточного программного обеспечения.

Уровни архитектуры веб-приложений

Уровень представления (PL) отображает пользовательский интерфейс и упрощает взаимодействие с пользователем

- Уровень обслуживания данных (DSL) передает данные, обработанные уровнем бизнес-логики, на уровень представления.
- Уровень бизнес-логики (BLL) несет ответственность за надлежащий обмен данными.
- Уровень доступа к данным (DAL) предлагает упрощенный доступ к данным, хранящимся в постоянных хранилищах.

Архитектуры веб-приложений:

- Одностраничные веб-приложения
- Многостраничные веб-приложения
- Архитектура микросервисов
- Бессерверная архитектура
- Прогрессивные веб-приложения

Основы архитектуры веб-приложений. Основные принципы:

- *Многоуровневая архитектура* делит приложение на разные логические уровни, каждый из которых несет определенную ответственность.
- *Разделение задач* — это фундаментальный принцип архитектуры веб-приложений, который включает разделение различных аспектов приложения на отдельные компоненты, такие как управление данными, пользовательский интерфейс и бизнес-логика.
- *Архитектура модель-представление-контроллер (MVC)* — это шаблон проектирования, разделяющий приложение на три компонента: модель (уровень данных), представление (пользовательский интерфейс) и контроллер (бизнес-логика).
- *Архитектура без сохранения состояния* обрабатывает каждый запрос к серверу как новый и независимый запрос, не полагаясь на какую-либо информацию о состоянии из предыдущих запросов.
- *Кэширование* предполагает хранение часто используемых данных в памяти, чтобы их можно было быстро извлечь без необходимости выполнения дорогостоящих запросов к базе данных.
- *Безопасность* является критическим аспектом архитектуры веб-приложений и включает в себя реализацию надлежащих мер для защиты приложения от распространенных угроз, таких как межсайтовый скриптинг (XSS), SQL-инъекция и сеанс угон.

Схема сайта — план-схема размещения его основных разделов и страниц относительно друг друга, которая показывает, каким образом строится сайт, логическая связка его страниц. Например: линейная, сложная линейная, блочная, древовидная.

Макет сайта — предварительный эскиз, определяющий не только внешний вид будущего ресурса, но и то, как пользователи будут с ним взаимодействовать. Макет —

основа, вокруг которой выстраивается концепция будущего сайта. Он удобен для всех сторон – и для команды разработчиков, и для заказчика:

1. Визуализация внешнего вида ресурса;
2. Предугадывание желаний;
3. Определение недочетов;
4. Оценка бюджета и поиск инвесторов.

Какие требования выдвигаются к готовому макету сайта

Симметричность. При создании макетов сайтов на начальном этапе дизайнеры рисуют простейшую схему, состоящую из вертикальных и горизонтальных линий.

UI/UX-дизайн. Макет сайта – это не только о красоте, нужно позаботиться о UI/UX-дизайне (пользовательский интерфейс и опыт соответственно). На стороне UX готовый дизайн должен быть логичным, простым для восприятия и не требующим потери времени на поиск объектов. UI отвечает за эстетику, гармонию и сочетаемость: компоновка перекликающихся оттенков, шрифтовые пары, оформление кнопок и многое другое.

Структура. Существуют базовые правила, которые используют верстальщики и веб-дизайнеры для обеспечения привычной для пользователей структуры.

- шапка сайта с меню (header).
- информационный раздел (body, тело)
- футер (footer или подвал).

Композиция:

- направление взгляда.
- якорные элементы для создания точки фокусировки.
- цветовые решения.

Адаптивность. Современные сайты создают с оглядкой на то, что пользователи будут посещать их с разных устройств: **смартфоны, ноутбуки, компьютеры, телевизоры** – под каждый экран дизайн должен подстраиваться.

Темы

1. Культура и искусство
2. Медицина
3. Образовательный сайт
4. Работа и рекрутинг
5. Стоимость билетов на транспорт до городов
6. Страны, язык страны, население стран
7. Врачебные услуги
8. Туристические поездки
9. Футбольные матчи
10. Спортивные состязания
11. Продажа квартир
12. Приложения для обучения
13. Приложения для тестирования

Задание

1. Выбрать архитектуру web-приложения. Аргументировать, почему эта архитектура лучше подходит для вашего приложения
2. Построить схему приложения

3. Построить макет клиентской части
4. Защитить лабораторную работу

Отчет должен содержать следующие разделы

1. Титульный лист
2. Введение (4-5 предложений)
3. Постановка задачи
4. Представить архитектуру web-приложения. Аргументация архитектуры, которую вы будете использовать
5. Схема сайта
6. Макет сайта
7. Сохранить файл отчёта по шаблону: Группа_ФамилияИО_ЛабРаб1

Например

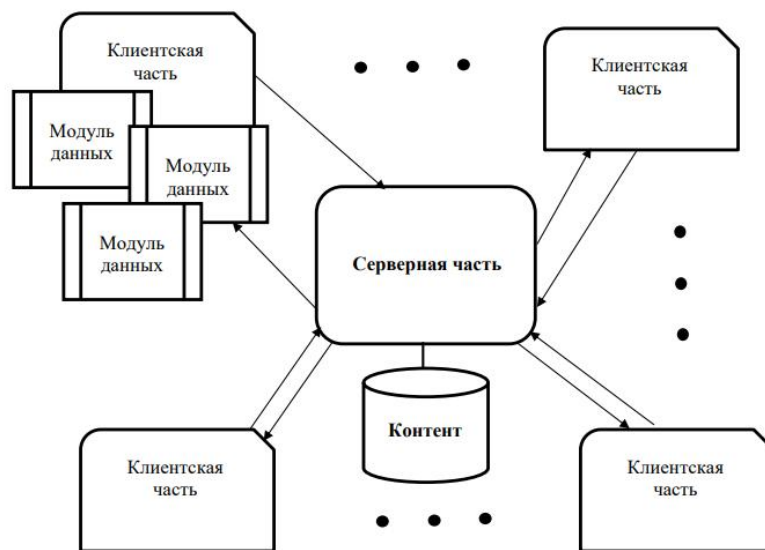


Рисунок 2.2- Типовая структура web приложения