

1. ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1.1. Этапы и сроки выполнения курсовой работы

Тематика курсовых работ (КР) представлена отдельным списком. Курсовая работа разрабатывается студентами очной формы обучения в сроки, предусмотренные графиком учебного процесса. Законченную курсовую работу (web-сервис и пояснительную записку) студент должен представить на кафедру до начала сессии. После проверки преподавателем, при условии отсутствия замечаний, требующих устранения, КР до защиты остается на кафедре. Если преподаватель указал замечания, требующие устранения, студент должен устранить их до защиты работы.

Общий объем работы 30–60 страниц машинописного текста (без приложений). Приложения формируются при необходимости, если имеющиеся таблицы, рисунки, расчеты затрудняют восприятие материала курсовой работы.

Структурными элементами курсовой работы являются:

- титульный лист;
- задание на курсовую работу;
- содержание;
- введение;
- основная часть;
- заключение;
- библиографический список;
- приложение.

Титульный лист является первой страницей курсовой работы, оформляется в соответствии с требованиями кафедры и не нумеруется.

Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование) основной части, заключение, библиографический список и приложения с указанием номеров страниц, с которых начинаются эти элементы работы

Введение должно содержать оценку современного состояния технологической области, с которой связано выполнение курсовой работы. При раскрытии содержания применяемых технологий, как правило, используются описания стека применяемых технологий, отдельных концепций и технологий. Указываются методы научного поиска, дается обзор использованных источников и литературы. Затем формулируется цель курсовой работы, а также комплекс взаимосвязанных задач, подлежащих решению.

Основная часть включает четыре раздела:

1. Теоретические основы предметной области.
2. Web-дизайн сервиса. Разработка базы данных.
3. Программная реализация web-сервиса.
4. Тестирование и публикация web-сервиса.

В зависимости от особенностей работы основную часть излагают в виде сочетания текста, таблиц, формул, иллюстраций и листингов программного кода разработанного сервиса. Основную часть следует делить на разделы, подразделы и пункты (при необходимости). Первый раздел работы носит теоретический характер. В нем рассматриваются платформы, а также их

инструментарии для создания web-сервиса. Описываются их сильные и слабые стороны. Обосновывается выбор одной из них, позволяющей эффективно реализовать поставленные задачи. Во втором разделе излагается web-дизайн сервиса или API если сервис имеет RESTful интерфейс, который состоит из нескольких этапов.

Первый этап создания сервиса – аналитический. На этом этапе анализируются задачи, которые следует учитывать при создании web-дизайна (API интерфейса), изучаются потенциальные клиенты web-сервиса. Затем клиенты сервиса делятся на различные категории, определяются цели и сценарии посещения сервиса каждой группой посетителей.

Второй этап. На основе данных аналитики, полученных на первом этапе работ, начинается разработка концепции и структуры ресурса – основа дальнейших работ по созданию web-дизайна и текстового наполнения сервиса. При разработке сервиса очень важным моментом является разработка правильной концепции. Концепция сервиса представляет собой совокупность обозначенных целей, сформулированных в емкой, интересной форме, удобной для изучения пользователем. Эффективная концепция – залог того, что создание сервиса в целом и web-дизайна в частности будет корректным с точки зрения предстоящих задач. Структура ложится в основу дизайна web-сервиса и должна учитывать потребности и предпочтения целевой аудитории, следовать ее поведенческим мотивациям и сценариям для каждой группы пользователей. Логически обоснованная структура – необходимый элемент, без которого невозможно разработать качественный дизайн и текстовое наполнение web-сервисов. Дизайн для web-сервисов, при разработке которого не учитываются логические связки в структуре, в дальнейшем имеет проблемы с количеством пользователей и качеством навигации.

Третий этап создания сервиса – непосредственно web-дизайн (API интерфейс). Предполагает соблюдение ряда обязательных требований:

1. Максимально удобный интерфейс: логика, структура и система навигации сервиса (url - ссылки).
2. Графическое решение: при этом web-дизайн сервиса должен быть разработан с ориентацией не на эстетические взгляды создателей, а на вкусовые предпочтения целевой аудитории.
3. Web-дизайн не должен мешать восприятию и читабельности текста.
4. Web-дизайн должен быть разработан таким образом, чтобы, несмотря на все красоты, не увеличивать скорость загрузки страниц сервиса. Без соблюдения этих правил невозможна разработка эффективного web-дизайна.

Третий раздел представляет собой описание программной реализации web-сервиса. В силу того, что описать весь сервис не представляется возможным, необходимо уделить внимание, прежде всего, описанию программной реализации ключевых элементов.

Четвертый раздел должен быть посвящен тестированию сервиса. В ходе тестирования проверяются программные модули, интерфейс, текстовое наполнение и web-дизайн, все те компоненты, от которых зависит эффективность работы ресурса. Создание сервисов – многоступенчатый и многокомпонентный процесс, и тестирование помогает проверить, все ли этапы были качественно реализованы. В случае выявления слабых мест, неудобных логических связей, программных сбоев проводится корректировка, меняется текст, web-дизайн и т. д. Успешное тестирование – необходимый показатель, без которого невозможно оценить, насколько грамотно было выполнено создание сервиса.

В **заключении** формулируются краткие выводы по проделанной работе, дается оценка степени выполнения поставленных задач, полученных проектных разработок.

Библиографический список должен содержать сведения об источниках, использованных при выполнении КР. В приложение рекомендуется включать материалы, связанные с выполнением курсовой работы (таблицы, графики, рисунки), если они затрудняют восприятие материала, а также листинг кода разрабатываемого web-сервиса.

2. ЦЕЛЬ РАБОТЫ

Целью работы является изучение основ создания динамических сервисов на платформе **Tomcat** (язык **Java**) либо **Django** (Языка **Python**). В процессе выполнения курсовой работы студенты должны разработать структуру web-сервиса (в зависимости от тематики структура сервиса может отличаться), наполнить его содержательной информацией, создать соответствующий программный модуль с использованием современных инструментальных средств, реализовать поиск данной информации в рамках сервиса. Вся ключевая информация должна храниться в базе данных **Postgres SQL** версии не ниже 9.5.

3. СОСТАВ И ПОСЛЕДОВАТЕЛЬНОСТЬ РАБОТ

Состав и последовательность работ следующие:

- изучение темы работы и назначения web-сервиса, предметной области;
- сбор содержательной информации по теме курсовой работы;
- анализ и выбор инструментальных средств реализации web-сервиса;
- разработка дизайн-макета главной и остальных страниц сервиса, (API интерфейса RESTful сервиса), решение вопросов о применении фреймворков структур, графических средствах, мультимедийных возможностях.
- разработка состава и структуры web-сервиса, разработка структуры базы данных, определение содержания каждой страницы, проектирование системы навигации (взаимосвязь информационных блоков и гиперссылок);
- определение порядка сопровождения и обновления страниц;
- создание web-сервиса с помощью выбранного инструментального средства, заполнение базы данных соответствующей информацией;
- разработка подсистемы поиска информации в рамках сервиса (с использованием запросов к базе данных);
- изучение исходного кода страниц, написание комментариев;
- публикация web-сервиса;
- тестирование web-сервиса;
- оформление курсовой работы в соответствии с подразделом
- защита работы с демонстрацией результатов работы

4. ТРЕБОВАНИЯ К WEB – СЕРВИСУ

К web-сервису предъявляются следующие требования:

1. Web-сервис должен быть выполнен в едином стиле.
2. Web-сервис должен корректно работать в браузерах Opera, Mozilla Firefox, Google Chrome и Safari.
3. Применяемые технологии клиентской части:
 - a. JavaScript

- b. CSS
 - c. HTML
 - d. jQuery
 - e. Bootstrap
 - f. Дополнительные технологии по согласованию с преподавателем.
4. Применяемые технологии серверной части
 - a. Java, Spring boot, Thymeleaf
 - b. Python, Django
 5. Время загрузки каждой страницы должно быть минимальным.
 6. Каждая страница должна содержать удобные и понятные средства навигации по сервису в различных направлениях.
 7. Применение фреймов, списков, таблиц, графики, средств мультимедиа должно быть разумным и соответствовать тематическому направлению сервиса.
 8. На сервисе может быть реализована возможность регистрации и авторизации пользователей, разграничение прав между гостевыми посетителями и зарегистрированными пользователями.
 9. Работоспособность web-сервиса должна быть протестирована одним из известных способов.

Этапы проектирования курсового проекта

Срок до	Шаг	Материалы
7 неделя	Перечень сущностей предметной области	Предоставить список сущностей предметной области выбранной темы
8 неделя	Модель БД. Данные	Предоставить модель БД, спроектированную для хранения доменной области, взаимосвязи между сущностями, наборы полей и тестовые данные для наполнения.
11 неделя	Проектирование серверной части	Предоставить проект серверной части приложения, с минимальным клиентским дизайном. В серверной части для объектной моделей должны быть спроектированы контроллеры, методы обработки запросов. Проект должен запускаться и отвечать на клиентские запросы
13 неделя	Проектирование клиентской части	Предоставить дизайн клиентской части, или API RESTful запросов. Клиентские запросы могут выполняться как асинхронно, так и синхронно
15 неделя	Оформление ПЗ	На данном этапе, возможно параллельно с предыдущими этапами формируется пояснительная записка, основные главы, наполнение
16 неделя	Защита	