МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ШКОЛА КОМПЬЮТЕРНЫХ НАУК

ОТЧЕТ

О РАЗРАБОТКЕ ПРОЕКТА ПО ДИСЦИПЛИНЕ

«Интеллектуальные системы поддержки принятия решений»

ФГАОУ ВО «Тюменский государственный университет»

(наименование организации)

Кафедра программного обеспечения

(наименование структурного подразделения)

|  |  |  |
| --- | --- | --- |
| Выполнил  обучающийся 4 курса,  МОиАИС-20.01 группы | (подпись) | Мирзаева Саодат  Синдар кизи |
| Научный руководитель проекта  Профессор, д-р физ.-мат. наук |  | Шевляков Артем Николаевич |

|  |  |  |
| --- | --- | --- |
| Принял защиту отчёта  Доцент, к. ф.-м. н. |  | Плотоненко Юрий Анатольевич |

Защищен 2024 г.

Результаты / зачета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись руководителя практики от института)

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc185588311)

[ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ. 5](#_Toc185588312)

[1.1 ПРЕДМЕТНАЯ ОБЛАСТЬ 5](#_Toc185588313)

[1.2 ОБЗОР АНАЛОГОВ 7](#_Toc185588314)

[1.3 ПОСТАНОВКА ЗАДАЧИ 9](#_Toc185588315)

[1.4 ИСПОЛЬЗУЕМЫЕ ПРИЕМЫ ВИЗУАЛИЗАЦИИ. 10](#_Toc185588316)

[1.5 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ 13](#_Toc185588317)

[ГЛАВА 2. РАЗРАБОТКА ПЛАТФОРМЫ С ОПИСАНИЕМ УЧЕБНЫХ ПРОЕКТОВ И ОТСЛЕЖИВАНИЕМ ИХ ВЫПОЛНЕНИЯ. 16](#_Toc185588318)

[2.1 РАБОТА С ДАННЫМИ 16](#_Toc185588319)

[2.2 АРХИТЕКТУРА ПЛАТФОРМЫ 20](#_Toc185588320)

[2.3 ОТСЛЕЖИВАНИЕ ВЫПОЛНЕНИЯ ПРОЕКТА. 22](#_Toc185588321)

[ГЛАВА 3. РЕЗУЛЬТАТЫ. 23](#_Toc185588322)

[3.1 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВЕБ-ПРОЕКТА 23](#_Toc185588323)

[3.2 АРХИТЕКТУРА ВЕБ-ПРОЕКТА 23](#_Toc185588324)

[3.3 СТРУКТУРА ВЕБ-ПРОЕКТА 24](#_Toc185588325)

[3.1 ФУНКЦИОНАЛ ПЛАТФОРМЫ. 29](#_Toc185588326)

[3.3 ПЛАН РАБОТ 37](#_Toc185588327)

[ЗАКЛЮЧЕНИЕ 38](#_Toc185588328)

[СПИСОК ЛИТЕРАТУРЫ 39](#_Toc185588329)

# ВВЕДЕНИЕ

В настоящее время образование играет ключевую роль в успехе каждого человека. Постоянно возрастающие требования к качеству образования и сокращение времени на его получение представляют серьезные проблемы для студентов и преподавателей. В связи с этим возникает необходимость в разработке инновационных решений, которые бы позволили оптимизировать процесс обучения и повысить его эффективность.

Целью данного проекта является разработка платформы, которая предоставит возможность создания и ведения учебных проектов с описанием и отслеживанием их выполнения. Основной пользователь данной платформы – студенты и преподаватели.

В настоящее время студенты и преподаватели сталкиваются со множеством трудностей при работе с учебными проектами. Отсутствие централизованной системы, объединяющей все проекты одного образовательного учреждения, затрудняет контроль и управление процессом обучения. Нередко преподаватели сталкиваются с проблемой отслеживания прогресса студентов, а студенты имеют сложности с организацией и документированием своей работы. Недостаточное взаимодействие между участниками проекта создает проблемы в коммуникации и сотрудничестве.

Для решения данной проблемы необходимо разработать платформу, которая бы объединяла всех участников проекта, обеспечивала удобную работу с учебными проектами. Данная работа предоставит возможность создания и редактирования проектов, добавления задач, определения сроков выполнения и отслеживания прогресса участников.

Цель проекта: создать MVP платформы для участников образовательного процесса (студентов и преподавателей), которая будет предоставлять возможность описания учебных проектов и отслеживания их выполнения.

В ходе работы над проектом необходимо решить следующие задачи:

1. Создать базу данных для хранения информации.
2. Разработать пользовательский интерфейс платформы.
3. Реализовать функционал платформы.
4. Найти модели для визуализации ведения и выполнения проекта.
5. Реализовать визуализацию прогресса участников платформы по проектам, задачам и подзадачам для отслеживания их выполнения.

В итоге, разработка данной платформы позволит осуществлять ведение учебных проектов в одном информационном пространстве. Участники проекта смогут легко и организованно работать, отслеживать свой прогресс и сотрудничать друг с другом, что сделает образование более интерактивным и привлекательным.

# ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.

## ПРЕДМЕТНАЯ ОБЛАСТЬ

Предметная область связана с образованием и с процессами, которые сопровождают ведение проектов.

Платформа предоставляет возможность студентам и преподавателям создавать, просматривать учебные проекты и отслеживать их выполнение.

Что такое платформа? В отличие от обычного сайта, который предоставляет только статическую информацию, платформа предоставляет возможность создания и участия в различных интерактивных событиях и функциях. Основное отличие в том, что платформа — это более сложная система, обеспечивающая интерактивное общение и сотрудничество пользователей, часто в реальном времени. На платформе пользователи могут общаться, делиться контентом, управлять различными процессами и проектами, проводить обучение и т.д., в то время как обычный сайт, как правило, предлагает только просмотр информации.

Учебные проекты – основное составляющее платформы. Каждый проект содержит описание, сроки начала и завершения выполнения, статус (В ожидании, в процессе, выполнено, отменено), автора, команду и задачи. Это могут быть различные проекты, связанные с программированием, дизайном, маркетингом и другими областями. В данном случае платформа больше направлена на IT-проекты.

Отслеживание выполнения: Система позволяет отслеживать прогресс выполнения учебных проектов в виде таблиц и инструментов визуализации.

Процессы, сопровождающие ведение проектов, включает в себя ряд деятельностей, которые необходимы для успешной реализации проекта.

Основные компоненты этих процессов включают:

Планирование проекта: определение целей, задач и ресурсов, необходимых для реализации проекта.

Управление ресурсами: распределение ресурсов, назначение задач, контроль времени.

Контроль выполнения задач: отслеживание хода выполнения работ, выявление и устранение задержек и проблем.

Анализ рисков: оценка возможных угроз проекту и разработка планов по их устранению.

Коммуникация с заинтересованными сторонами: поддержание связи с участниками проекта, учет их интересов и требований.

Оценка результатов: анализ полученных результатов, выявление проблем и ошибок, работа над улучшением процессов.

Всего есть две роли пользователей – студент и преподаватель. Роль пользователя определяется самим пользователем при регистрации на платформе. Преподаватель определяет команду студентов и прикрепляет ее к ранее созданному проекту, отслеживает выполнение с помощью дашборда. Студенту доступны только те проекты, к которым он прикреплен. Студент может отслеживать свое время работы над проектом.

## ОБЗОР АНАЛОГОВ

Существует множество аналогов нашей платформы, но у все этих решений есть свои плюсы и минусы, а также различные проблемы и цели.

Кейсы БММ ТюмГУ.

Данная платформа нацелена на решение проблем предпринимателей. Создается кейс с проблематикой и набирается команда для реализации данного проекта.

В отличии от данной платформы наша решает проблему ведения и отслеживания проектов внутри образовательного процесса. Темы проектов студенты придумывают самостоятельно. Проекты не обязательно коммерциализировать.

СИБТЕХ. Витрина проектов.

Возможности сайта:

* Размещение каталога проектов и отзывов участников;
* Подача заявки на реализацию проекта;
* Заполнение формы на участие в проекте;
* Выбор команды проекта на Витрине участников.

Функции CRM-системы:

* Доработка заявок прямо из системы, согласование проекта с коллегами;
* Автоматическая публикация на сайте одобренных проектов;
* Проверка анкет участников на правильность заполнения;
* Автоматическая выгрузка анкет на Витрину участников из CRM-системы;
* Автоматическая генерация и отправка сертификата участникам;
* Единая платформа для работы над проектом.

Данная платформа нацелена на отслеживание выполнения проектов, но здесь нет взаимодействия «преподаватель – проект». Также данная платформа платно создается индивидуально для заказчиков в лице СУЗов, ВУЗов.

## 1.3 ПОСТАНОВКА ЗАДАЧИ

1. Постановка задач для всего проекта:

Дано:

Студенты, которые обучаются на разных направлениях, в разных группах.

Проекты, которые реализуются студентами либо в рамках образовательного процесса, либо по инициативе студентов.

Преподаватели, которые хотят видеть, как протекает работа над проектами, которые реализуются в рамках образовательного процесса.

Задача:

Организовать пространство, где студенты смогут выполнять работу над своими проектами в командах, а преподаватели смогут отслеживать время выполнения проекта по каждому студенту.

2. Постановка задачи для отслеживания выполнения проекта.

Дано:

{pi}, i = 1, …, n - множество проектов, pi P.

Проект pi представлен кортежем: [starti, finishi, statusi,{}, m=(1, …,j), , k= (1, …, l)], где: starti, finishi – дата начала и окончания выполнения проекта, statusi – статус выполнения проекта, – другие атрибуты проекта (название, описание, и др.), {} – множество задач проекта.

Задача представлена кортежем: [namem, statusm, {}, g = (1, …, s), , k= (1, …, r)], где: namem – название задачи, statusm – статус выполнения задачи, – другие атрибуты задачи (описание и др.), {} - множество прогрессов для задачи.

## 1.4 ИСПОЛЬЗУЕМЫЕ ПРИЕМЫ ВИЗУАЛИЗАЦИИ.

Графическая схема взаимосвязи цели, подцелей и задач по их

достижению именуется «Дерево целей проекта». Эта структура появляется в результате декомпозиции цели проекта (рис. 1).

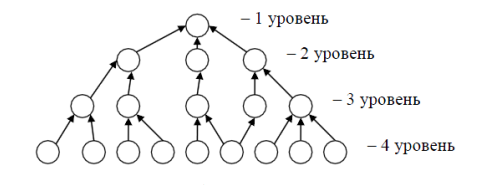


Рис. 1 Дерево целей проекта

Представление целей начинается с верхнего уровня, дальше они последовательно разукрупняются. При этом основным правилом разукрупнения целей является полнота: каждая цель верхнего уровня должна быть представлена в виде подцелей следующего уровня исчерпывающим образом. Согласно одному из наиболее распространенных подходов к определению целей проекта (SMART), они должны быть:

- конкретными (Specific), определяющими, что должно быть достигнуто и к какому сроку;

- измеримыми (Measurable) посредством цены, качественных и количественных параметров;

- достижимыми (Attainable) в пределах знаний, опыта, интенсивности потребления ресурсов и т. п.;

- реалистичными (Realistic), т. е. достижимыми, но требующими усилий;

- контролируемыми (Trackable), согласованными по датам и методам измерения достигнутого успеха

Принципы декомпозиции цели проекта:

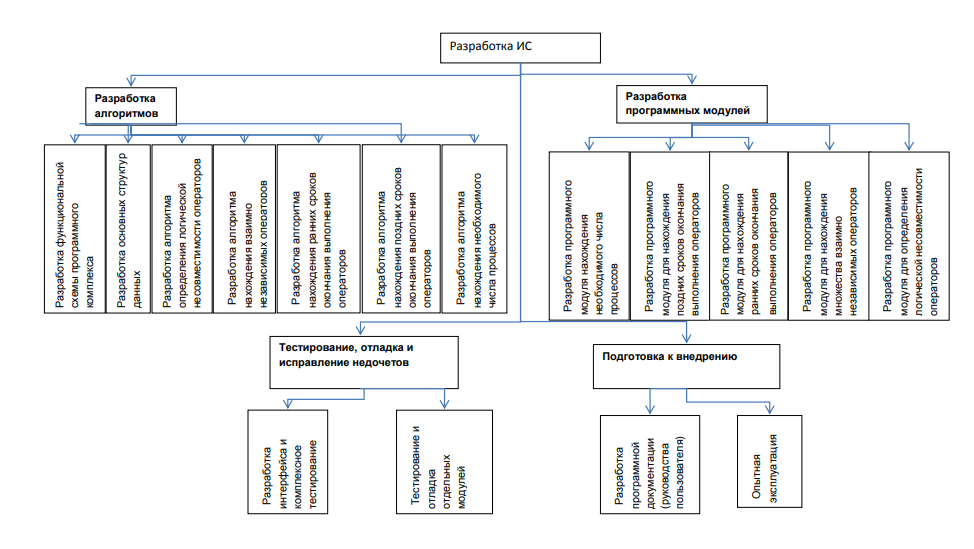
1. Реализация подцелей каждого последующего уровня является необходимым и достаточным условием достижения цели предыдущего уровня. Нумерация уровней идет сверху вниз.

2. В структуре указываются желаемые результаты проекта и его частей, а не способы их получения.

3. Подцели каждого уровня должны быть независимы друг от друга и не выводимы друг из друга.

4. Фундамент «дерева целей» должны составлять задачи, которым соответствуют элементарные результаты проекта.

5. Детализировать деятельность полезно до уровня, обеспечивающего возможность получения отчетности, необходимой для руководителя проекта и заказчика. В результате декомпозиции появляется перечень работ по проекту. Пример декомпозиции цели проекта. В качестве основы для проведения декомпозиции цели проекта можно использовать этапы жизненного цикла ИС, приведенные в приложении 1.



Приложение 1. Этапы жизненного цикла ИС

Канбан - доска. Цель системы канбан — сделать проект наглядным, отследить готовность работ и проконтролировать нагрузку специалистов. Канбан доска используется как способ визуализации статусов проектов, задач и подзадач, потому что она позволяет участникам команды быстро и наглядно видеть, какие задачи находятся в работе, какие завершены и какие еще предстоит выполнить. Это помогает улучшить коммуникацию и координацию внутри команды, ускоряет процесс принятия решений и оптимизирует рабочий процесс. Канбан состоит из колонок, обозначающих различные стадии выполнения задачи ("В ожидании", "В процессе", "Выполнено"). Каждая задача представлена отдельной карточкой, которая перемещается по колонкам в соответствии с ее текущим статусом. Таким образом, участники команды всегда видят, какие задачи находятся в процессе выполнения, какие имеют приоритет, и могут легко определить, на каких задачах нужно сконцентрировать свое внимание.

Диаграмма Ганта - это тип гистограммы, которая иллюстрирует график выполнения проекта. На вертикальной оси перечислены задачи, которые необходимо выполнить, и временные интервалы на горизонтальной оси. Ширина горизонтальных полос на графике показывает продолжительность каждого действия. Диаграмма Ганта состоит из полос, ориентированных вдоль оси времени. Каждая полоса на диаграмме представляет отдельную задачу в составе проекта (вид работы), её концы — моменты начала и завершения работы, её протяженность — длительность работы. Каждая задача выделяется цветом, который соответствует определенному студенту, выполняющему задачу, что можно увидеть в легенде к гистограмме.

Круговая диаграмма представляет собой круговой [статистический график](https://en.wikipedia.org/wiki/Statistical_graphics), разделенный на фрагменты для иллюстрации числовых пропорций. На круговой диаграмме [длина дуги](https://en.wikipedia.org/wiki/Arc_length) каждого среза [пропорциональна](https://en.wikipedia.org/wiki/Proportionality_(mathematics)) величине, которую он представляет. В нашем случает такая диаграмма используется для визуализации всех проектов, задач и подзадач, цветом обозначается статус объекта представления.

Столбчатая диаграмма — диаграмма, представленная прямоугольными зонами (столбцами), высоты или длины которых пропорциональны величинам, которые они отображают. Прямоугольные зоны могут быть расположены вертикально или горизонтально. В нашем случае данный вид диаграммы показывает в каком месяце сколько проектов было создано.

Все вышеперечисленные средства позволяют управлять потоком работы, минимизировать потери времени на переключение между задачами и оптимизировать процесс достижения целей.

Метод критического пути (CPM) — это метод расписания, который определяет количество свободного времени или гибкость расписания для каждого из путей сети путем расчета самой ранней даты начала, самой ранней даты окончания, самой поздней даты начала и самой поздней даты окончания для каждого действия. Это метод сети расписания, который основан на последовательных сетях (одно действие происходит перед следующим, ряд действий, происходящих одновременно, завершается до начала следующего ряда действий и т. д.) и на единой оценке длительности для каждого действия. Метод диаграммы предшествования (PDM) может использоваться для выполнения CPM. Критический путь (CP) — это, как правило, самый длинный полный путь в проекте. Любая деятельность проекта с временем свободного времени, равным нулю, считается задачей критического пути. Критический путь может меняться при некоторых условиях. Время свободного времени также называется временем резерва.

## 1.5 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

При реализации проекта использовались следующие технологии.

В качестве основного языка программирование серверной части был выбран Python из-за его простоты, читаемости и богатой экосистемы библиотек и фреймворков. Кроме того, Python обладает хорошей масштабируемостью и производительностью, что делает его отличным вариантом для разработки веб-приложения. Был использован фреймворк Django для организации веб-сервера и обработки запросов от клиентов.

Также использован язык программирования Javascript для написания frontend и backend частей сайта. К нему были использованы Webpack инструмент, позволяющий скомпилировать JavaScript модули в единый JS-файл и кроссплатформенная библиотека Jquey используемая для упрощения клиентского сценария HTML. Также использовалась технология Ajax для обмена данными между веб-сервером и веб-страницей без необходимости перезагружать всю страницу.

Для разработки интерфейса веб-приложения использовались HTML, СSS и SCSS с фреймворком Bootstrap.

HTML – это стандартизированный язык разметки веб-страниц, SCSS, используется для описания внешнего вида сайта. Bootstrap – набор шаблонов для создания интерфейса веб-приложений. Последние две технологии из вышеперечисленных были выбраны из-за своей лёгкости в использовании, так как не пришлось бы расписывать стили элементов HTML-страниц с нуля, а только их видоизменять.

Bootstrap - свободный фреймворк, используемый для быстрой вёрстки адаптивного дизайна сайта и включающий в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения;

Visual Studio Code - редактор кода, используемый при работе с сервером.

Язык запросов дает несомненные преимущества, он продолжает идеологию архитектуры Клиент-Сервер. Клиентская часть веб-приложения подготавливает запрос на обработку информации и отсылает запрос на веб-сервер базы данных. Сервер, выполнив полученный запрос возвращает клиенту готовый результат.

Для реализации баз данных используется СУБД PostgreSQL. Основными достоинствами данной СУБД являются:

- Производительность: PostgreSQL обеспечивает высокую производительность при работе с различными типами данных и запросами. Это особенно важно для веб-приложений, где требуется быстрый доступ к данным и обработка большого количества запросов.

- Надежность и стабильность: PostgreSQL известен своей надежностью и устойчивостью к сбоям. Это также важно для веб-приложений, где каждая секунда работы системы имеет значение для пользователей.

- Расширяемость: PostgreSQL обладает богатым набором функций и возможностей, которые позволяют легко масштабировать и расширять базу данных по мере роста веб-приложения.

- Поддержка стандартов: PostgreSQL полностью соответствует стандартам ANSI SQL, что обеспечивает совместимость с другими базами данных и упрощает адаптацию к различным технологиям.

- Свободное и открытое ПО: PostgreSQL является свободно распространяемой и открытой СУБД, что обеспечивает нам возможность использовать её без лицензионных ограничений и снижает общие затраты на разработку.

# ГЛАВА 2. РАЗРАБОТКА ПЛАТФОРМЫ С ОПИСАНИЕМ УЧЕБНЫХ ПРОЕКТОВ И ОТСЛЕЖИВАНИЕМ ИХ ВЫПОЛНЕНИЯ.

## 2.1 РАБОТА С ДАННЫМИ

База данных состоит из 15 таблиц: django\_admin\_log, django\_content\_type, django\_migrations, django\_session, django\_migrations и auth\_group, Project, project\_module, project\_task, project\_team, project\_team\_user, schedule, schedule\_users, user, Связи между таблицами представлены на рисунке 2.1.

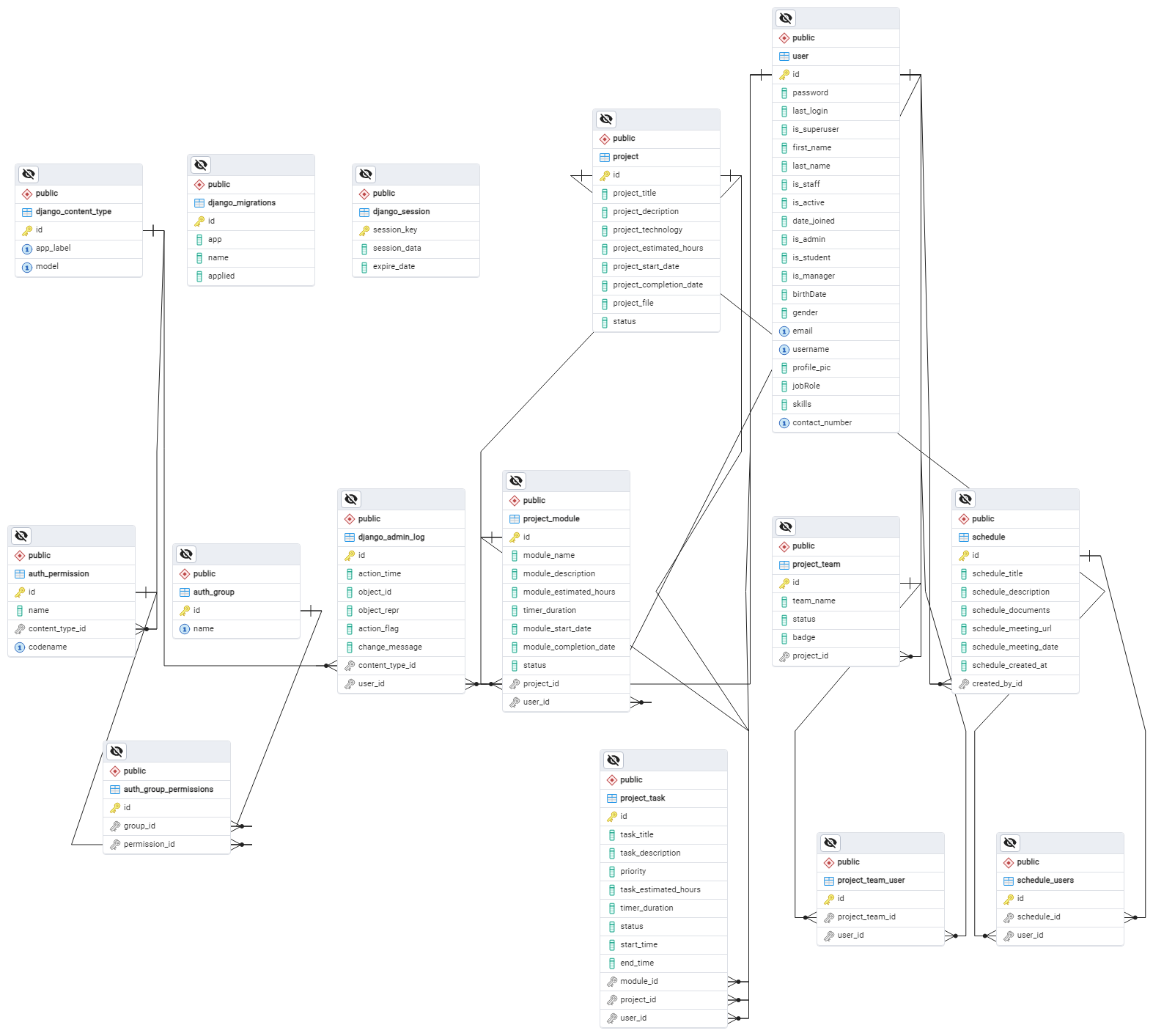


Рис. 2.1 Схема база данных

User– хранятся сведения о пользователях ( таблица 2.2).

Таблица 2.2. Описание таблицы User

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Password | Character varying | Пароль |
| Last\_login | Timestamp with zone | Логин |
| Is\_superuser | Boolean |  |
| First\_name | Character varying | Имя |
| Last\_name | Character varying | Фамилия |
| Is\_staff | Boolean | Персонал |
| Is\_active | Boolean | Активность |
| Date\_joined | Date | Дата присоединения |
| Is\_admin | Boolean | Админ |
| Is\_student | Boolean | Студент |
| Is\_teacher | Boolean | Преподаватель |
| Birthdate | Date | Дата рожения |
| Gender | Character varying | Пол |
| Email | Character varying | Почта |
| Username | Character varying | Имя пользователя |
| Profil\_job | Character varying | Профессия профиля |
| Job\_role | Character varying |  |
| Skills | Character varying | Навыки |
| Contact\_number | Character varying | Номер телефона |

Schedule– хранятся сведения о назначенных встречах( таблица 2.3).

Таблица 2.3. Описание таблицы Schedule

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Schedule\_title | Character varying | Название расписаний |
| Schedule\_description | Text | Описание расписаний |
| Schedule\_documents | Character varying | Документы расписаний |
| Schedule\_meeting\_url | Character varying | Ссылка расписаний встреч |
| Schedule\_meeting\_date | Date | Дата расписаний встреч |
| Schedule\_created\_at | Timestamp with zone | Дата расписаний |
| Created\_by\_id | Character varying | Создано идентификатором |

Project– хранятся сведения о проектах ( таблица 2.4).

Таблица 2.4. Описание таблицы Project

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Project\_title | Text | Название проекта |
| Project\_desription | Text | Описание проекта |
| Project\_technology | Character varying | Технологии проекта |
| Project\_estimated\_hours | Integer | Время отведенное проекту |
| Project\_start\_date | Date | Время начала проекта |
| Project\_file | Character varying | Файл проекта |
| Status | Character varying | Статус |

Schedule\_users – хранятся сведения о пользователях назначенных встреч студентов ( таблица 2.5).

Таблица 2.5. Описание таблицы Schedule\_users

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Schedule\_id | Bigint | Идентификатор расписания |
| User\_id | Bigint | Идентификатор пользователя |

Project\_team – хранятся сведения о командных проектах (таблица 2.6).

Таблица 2.6. Описание таблицы Project\_team

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Team\_name | Character varying | Название команды |
| Status | Character varying | Статус |
| Badge | Character varying |  |
| Project\_id | Bigint | Идентификатор проекта |

Project\_task – хранятся сведения о подзадачах проектов (таблица 2.7).

Таблица 2.7. Описание таблицы Project\_task

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Task\_title | Character varying | Название подзадачи |
| Task\_description | Text | Описание подзадач |
| Priority | Character varying | Приоритет |
| Task\_estimated\_hours | Integer | Время отведённые подзадачам |
| Timer\_duration | Interval | Отведенное время |
| Status | Character varying | Статус |
| Start\_time | Timestamp with zone | Время начала |
| End\_time | Timestamp with zone | Время сдачи |
| Module\_id | Bigint | Идентификатор задачи |
| Project\_id | Bigint | Идентификатор проекта |
| User\_id | Bigint | Идентификатор пользователя |

Project\_module – хранятся сведения о задачах проектов (таблица 2.8).

Таблица 2.8. Описание таблицы Project\_module

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Id | Bigint | Поле идентификатора |
| Module\_name | Character varying | Название задач |
| Module\_description | Text | Описание задач |
| Module\_estimated\_hours | Integer | Время отведённые задач |
| Module\_start\_date | Timestamp with zone | Дата начало |
| Module\_completiom\_date | Timestamp with zone | Дата завершение |
| Status | Character varying | Статус |
| Project\_id | Bigint | Поле идентификатора проекта |
| User\_id | Bigint | Ид пользователя |

## 2.2 АРХИТЕКТУРА ПЛАТФОРМЫ

В проекте используется клиент-серверная архитектура. Данная архитектура предоставляет централизованное хранение данных и доступ к ним через пользовательский интерфейс, обеспечивая удобство использования и масштабируемость системы (рис. 2.2).

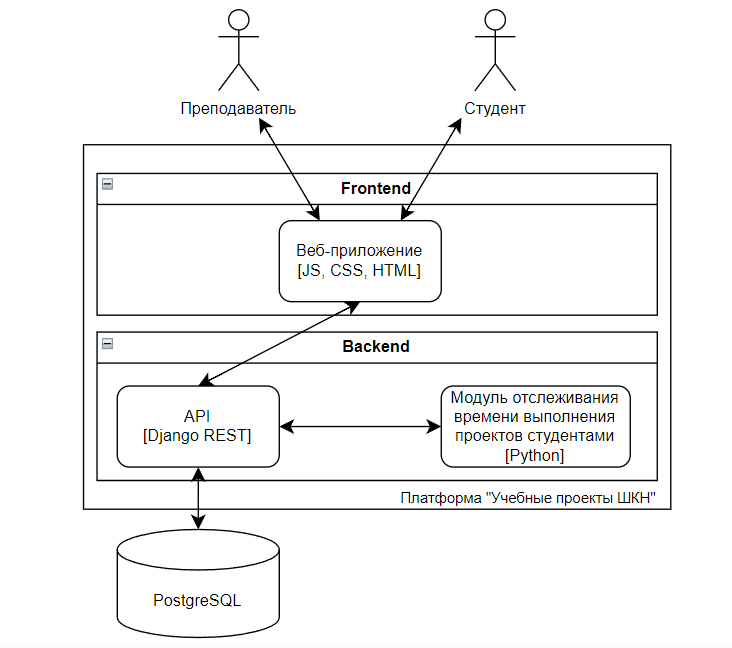


рис. 2.2 Архитектура платформы.

Клиентская часть:

Клиентская часть представляет из себя веб-интерфейс, через который пользователи могут просматривать информацию о проектах, вносить изменения в проекты, отслеживать прогресс выполнения проектов и т.д. Для обмена данными с сервером, клиентская часть использует HTTP-запросы к API сервера.

Серверная часть:

Сервер обеспечивает логику бизнес-процессов, связанных с управлением проектами, такие как добавление, изменение и удаление проектов, отслеживание прогресса выполнения проектов, уведомление пользователей об изменениях и т.д. Для взаимодействия с клиентами, серверная часть предоставляет API (application programming interface) для передачи данных и выполнения операций с учебными проектами.

В базе данных хранятся данных о проектах, пользователях, их выполнении и других связанных с учебными проектами информационных данных.

## 2.3 ОТСЛЕЖИВАНИЕ ВЫПОЛНЕНИЯ ПРОЕКТА.

На данный момент для отслеживания используется несколько приемов:

1. Дашборд для преподавателя, в котором используются разные графики для отображения информации.
2. Статусы выполнения для проектов, задач и подзадач – в виде доски канбан.
3. Диаграммы для подсчета всех проектов, задач, подзадач.
4. Графики для определения времени работы студентов над задачами и подзадачами.
5. Диаграмма Ганта – для визуализации сроков работ над каждой задачей проекта студентами.

# ГЛАВА 3. РЕЗУЛЬТАТЫ.

## 3.1 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВЕБ-ПРОЕКТА

Концепция разрабатываемого программного продукта представляет собой

веб-проект, построенный на веб-фреймворке Django, предоставляющий

следующие возможности:

* Отображение проектов на главной странице;
* Отображение задач и встреч на главной странице;
* Отображение проектов, включающих в себя описание задач и состав команд;
* Отображение форм регистрации и авторизации;
* Отображение Канбан доска;
* Отображение диаграмма Ганта
* Отображение Время выполнения задач и подзадач;
* Отображение список студентов;
* Отображение студенты в каком проекте участвует;
* Отображение назначенные встречи;
* Отображение статуса.

## 3.2 АРХИТЕКТУРА ВЕБ-ПРОЕКТА

Проект состоит из двух частей:

Общедоступная часть, которая позволяет пользователь создает проект в своем личном кабинете.

Административная часть, которая позволяет пользователем заполнять и редактировать базу данных сайта, а также отслеживать проекты и просматривать действие пользователей.

Эти две части реализованы, следуя шаблону MVT. Примем, что MVT

определяет способ разработки программного обеспечения, при котором код для

определения и доступа к данным (модель, файл models.py) отделен от логики

приложения (управление, файл views.py), которая в свою очередь отделена от

интерфейса пользователя (представление, файл html).

При создании веб-проекта на Django происходит разделение на сам веб проект и веб-приложение, которое создается после создания проекта.

## 3.3 СТРУКТУРА ВЕБ-ПРОЕКТА

В основу проекта входят:

Внешний каталог DjangoProject/ - это просто контейнер для проекта. Его

название никак не использует Django, и его можно переименовать во что

угодно.

Внутренний каталог DjangoProject/ - это пакет Python проекта. Его

название - это название пакета Python, которое будет использоваться для импорта чего-либо из проекта.

Во внутреннем находятся каталоги:

1. Каталог Project – это веб-приложение проекта;

2. Каталог Media – каталог, в котором находятся изображения, используемые в шаблонах веб-проекта;

3. Каталог Static – каталог, в котором находятся статические файлы и скрипты, используемые в веб-проекте.

Файл manage.py, находящийся в каталоге DjangoProject –это скрипт, который позволяет взаимодействовать с проектом Django.

Также во внутреннем каталоге DjangoProject создается каталог с одноименным названием содержащий файлы:

Файл settings.py – это файл, содержащий конфигурации проекта;

3.3.4 Структура веб-приложения

В веб-приложение входят:

Каталог Project – это каталог, в котором находятся файлы и каталоги приложения. В каталог Project входит каталог Migrations. Это каталог, который содержит все файлы миграций. Django создаст миграции при любых изменениях модели или полей – даже тех параметров, которые не влияют на базу данных – так как единственный способ восстановить состояние моделей это хранить все изменения в истории. «Начальные миграции» приложения – это миграции, которые создают первую версию таблиц для приложения. Обычно приложение содержит только одну начальную миграцию, но для сложной структуры моделей их может быть несколько. «Начальные миграции» помечаются атрибутом initial = True в классе миграции. Если атрибут initial не указан, миграция будет считаться «начальной», если это первая миграция в приложении (то есть она не зависит от другой миграции текущего приложения). Также в каталог Project входит каталог templates в котором хранятся шаблоны веб-приложения; models.py - файл отвечающий за модели в Django. Модели отображают информацию о данных. Они содержат поля и поведение данных. Обычно одна модель представляет одну таблицу в базе данных. В данной работе было сделано шесть моделей. Каждая модель представлена классом, унаследованным от django.db.models.Model. Каждая модель содержит несколько атрибутов, каждый из которых отображает поле в таблице базы данных. Каждое поле представлено экземпляром класса Field – например, CharField для текстовых полей. Это указывает Django какие типы данных хранят эти поля.

Названия каждого экземпляра Field - это название поля, в ―машинном‖

(machine-friendly) формате. Эти названия будут использоваться в коде, а база данных будет использовать их как названия колонок. Некоторые классы, унаследованные от Field, имеют обязательные аргументы. Например, CharField требует, чтобы эму передали max\_length. Это используется не только в схеме базы данных, но и при валидации. Также в моделях присутствует необязательный аргумент verbouse\_name = ‖ ‖ - это читабельное название модели, в единственном числе. Если этот параметр не указан, то в административной части отобразится имя поля. Дополнительные настройки для модели можно определить через class Meta, например название в единственной и множественной форме (verbose\_name и verbose\_name\_plural). Все они не обязательны и добавлять class Meta тоже не обязательно.

Также в некоторых моделях представлены два метода \_\_str\_\_ и get\_absolute\_url. Метод \_\_str\_\_ определяет название модели в административной части сайта. В данном примере в качестве названия берется название категории проекты. Метод get\_absolute\_url() необходим, чтобы указать Django как вычислить URL для объекта. Метод должен вернуть строку, которая может быть использована в HTTP запросе, функция image\_folder, необходимая для переименования картинок (в качестве нового названия берется slug модели), функция add\_to\_task, необходимая для добавления задач в базу данных, функция remove\_to\_task, необходимая для удаления зада из базы данных, функция change\_qty, необходимая для изменения количества человек, которые в проекте участвует.

admin.py – файл необходимый для использования моделей в административной части проекта. Код файла admin.py указан на рисунке 3.

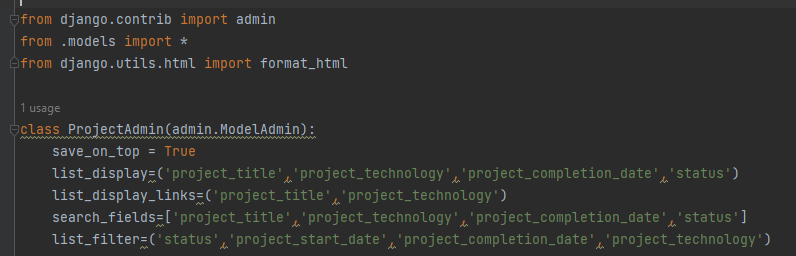


Рис. 3 Код файла admin.py

В файл импортируются все используемые модели и с помощью команды admin.site.register регистрируются для отображения в административной части;

forms.py – файл необходимый для работы с формами в общедоступной части. В файле описаны формы для регистрации, входа на сайт и форма для

проекта;

views.py – файл который отвечает за отображение общедоступной части. По сути это логика приложения. Отображение является функцией, которая обрабатывает HTTP-запрос, получает данные из базы данных, которые применяются для генерации страницы HTML. Затем функция отображения

возвращает сгенерированную страницу пользователю в виде HTTP-ответа;

urls.py – файл отвечающий за связь с файлом urls.py, находящимся в

основе проекта. Функция url() определяет URL-паттерн и функцию

отображения, которая будет вызвана, если введенный адрес будет соответствуют данному паттерну.(рис.3.1)

Данная функция url(), кроме того, определяет параметр name, который

уникально определяет это частное URL-преобразование.

apps.py – файл который определяет конфигурацию приложения.

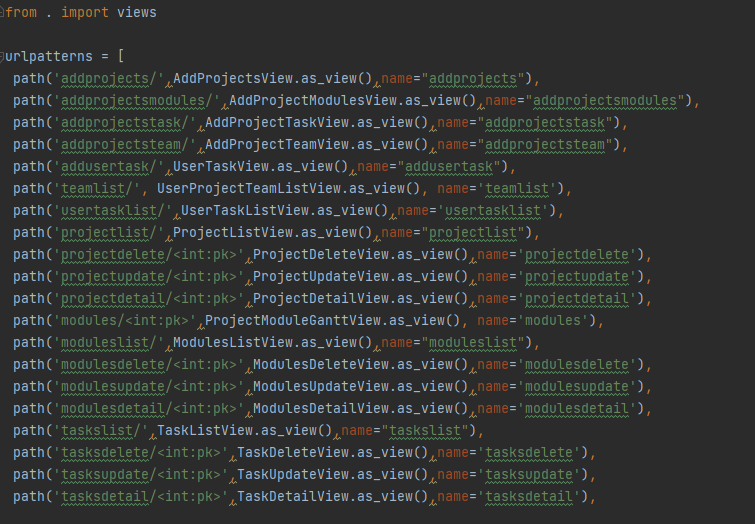


Рис. 3.1 Код файла url.py

Django содержит реестр установленных приложений, который содержит текущие настройки и предоставляет интроспекцию. Также предоставляет список доступных моделей. Этот реестр называется просто apps и находится в модуле django.apps. Полная схема работы веб-приложения представлена на рисунке 3.2.

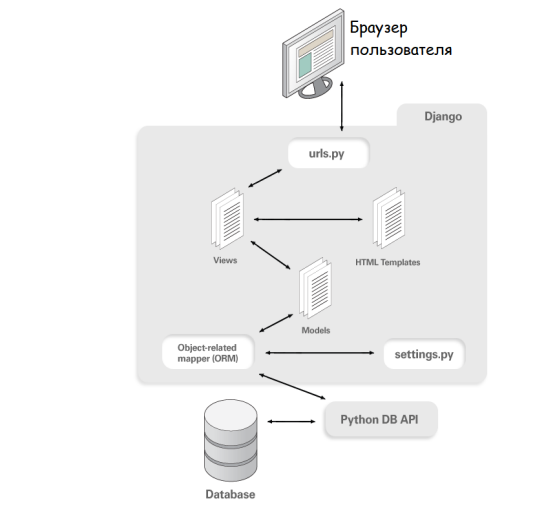


Рис. 3.2 Схема работы веб приложения

## 3.1 ФУНКЦИОНАЛ ПЛАТФОРМЫ.

На первой странице платформе необходимо ввести имя пользователя и его пароль, а также на платформе можно авторизоваться или зарегистрироваться (рис. 3.3). У администратора и преподавателя есть функционал для добавления пользователей на платформу.

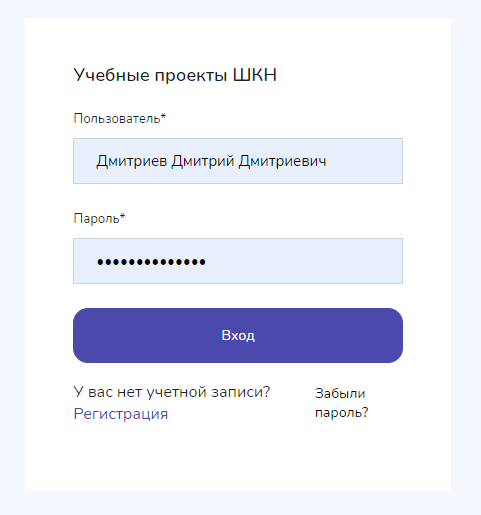


Рис. 3.3 Авторизация.

Авторизовавшись, пользователь попадает на главную страницу платформы, где представлены все проекты. Для студента и преподавателя доступны только проекты, в которых они учувствуют. Для администратора доступны все проекты платформы.

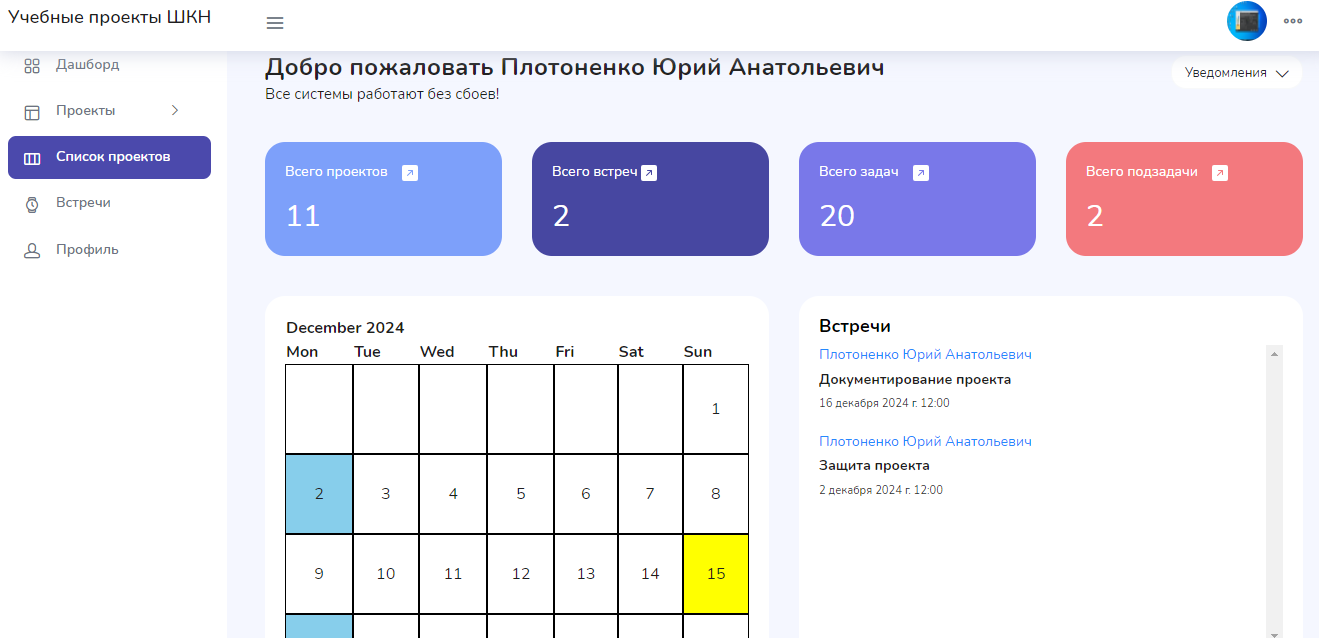
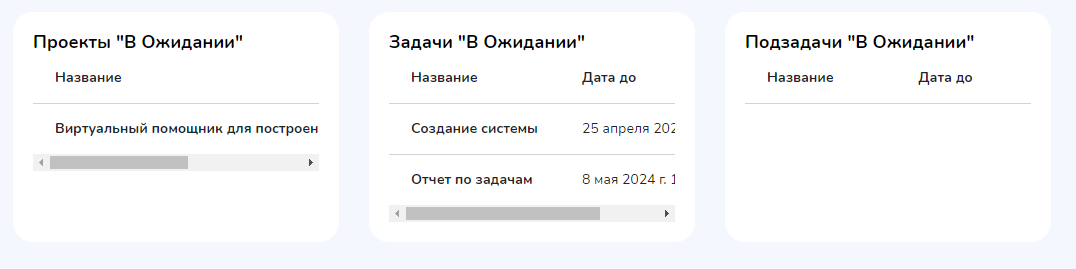
В дашборде преподавателя можно увидеть общее количество проектов, общее количество встреч, общее количество задач, а также общее количество подзадач. Кроме того здесь также отображается календарь в котором указаны даты всех предстоящих и прошедших встреч (рис. 3.4).

Рис. 3.4 Главная страница платформы.

В Канбан доске отображаются завершенные и не завершенные проекты, задачи и подзадачи тем самым увеличивая эффективность их выполнения (рис. 3.5).



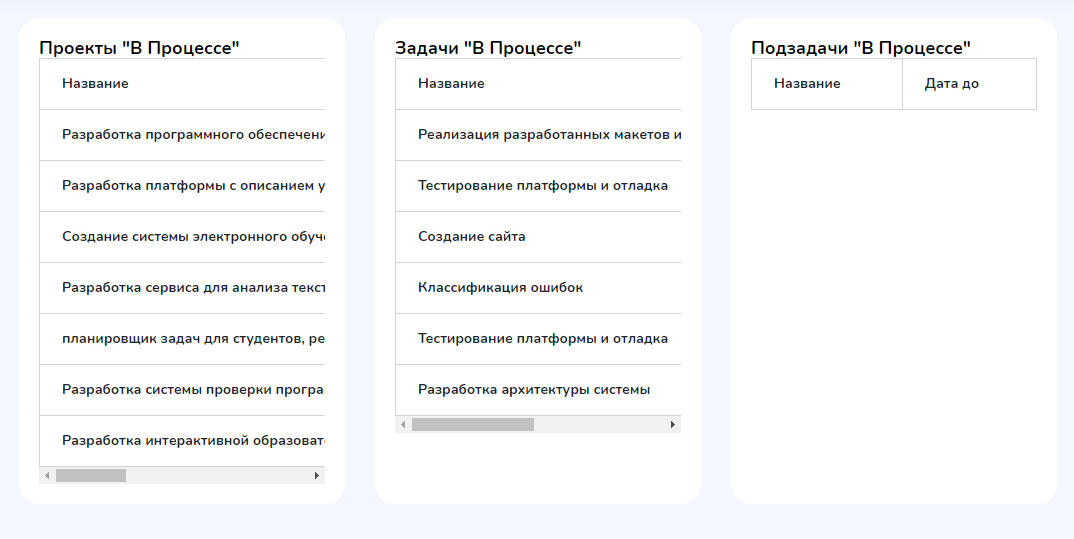
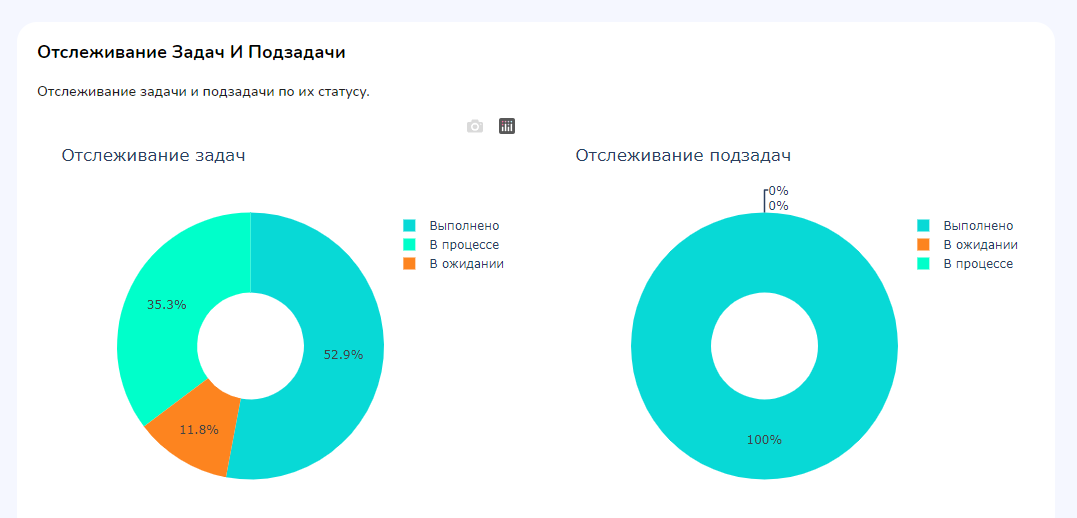


Рис.3.5 Канбан доска преподавателей

Диаграмма в процентах показывает общее количество выполненных задач, задач в процессе, а также общее количество задач в ожидании (рис.3.6). На этой странице отображается общее количество затраченных студентом времени на проект.



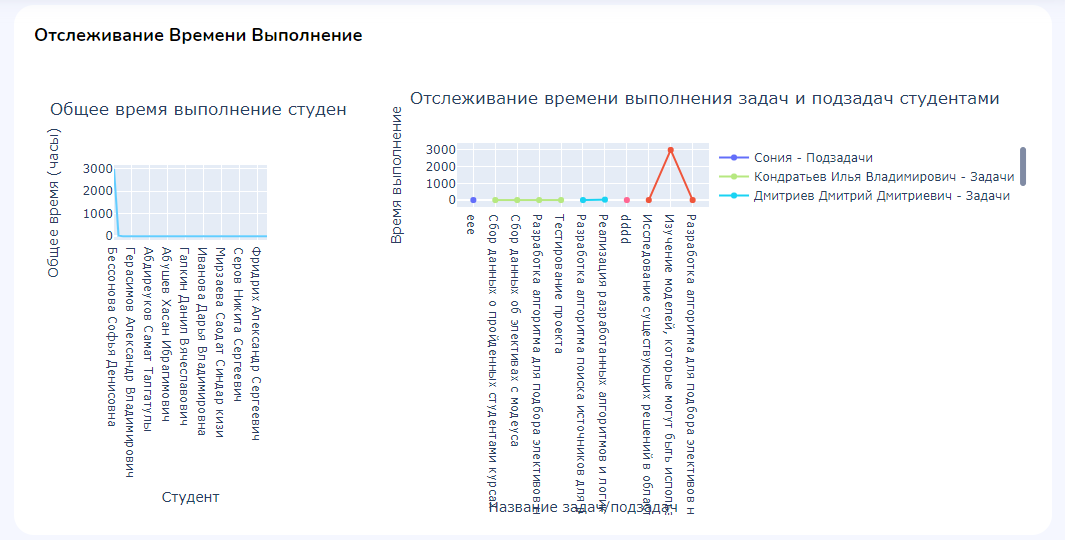


Рис. 3.6 Отслеживание задач и подзадач

На этой странице преподаватели могут увидеть общее количество проектов их студентов в таких статусах как: В Процессе, Выполнено, В Ожидании. А также преподаватель может просмотреть здесь количество созданных проектов за каждый месяц в виде гистограммы (рис. 3.7).

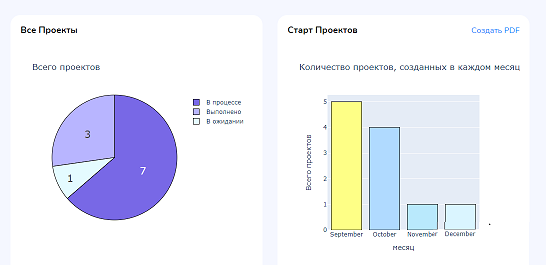


Рис. 3.7 Статус всего проекта

Во вкладке «Проекты» можно создать новый проект или посмотреть уже созданные проекты (рис 3.8). Во вкладке проекты доступны такие функции как Добавление Проектов, Добавление Задач, Добавление Подзадач, а также Создание Команды. Также в этой вкладке можно просмотреть уже созданные Проекты, Профиль и Намеченные Встречи.

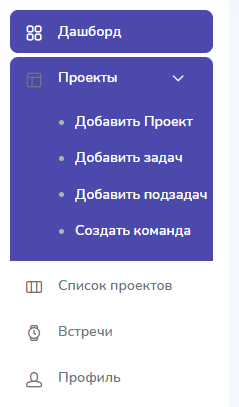
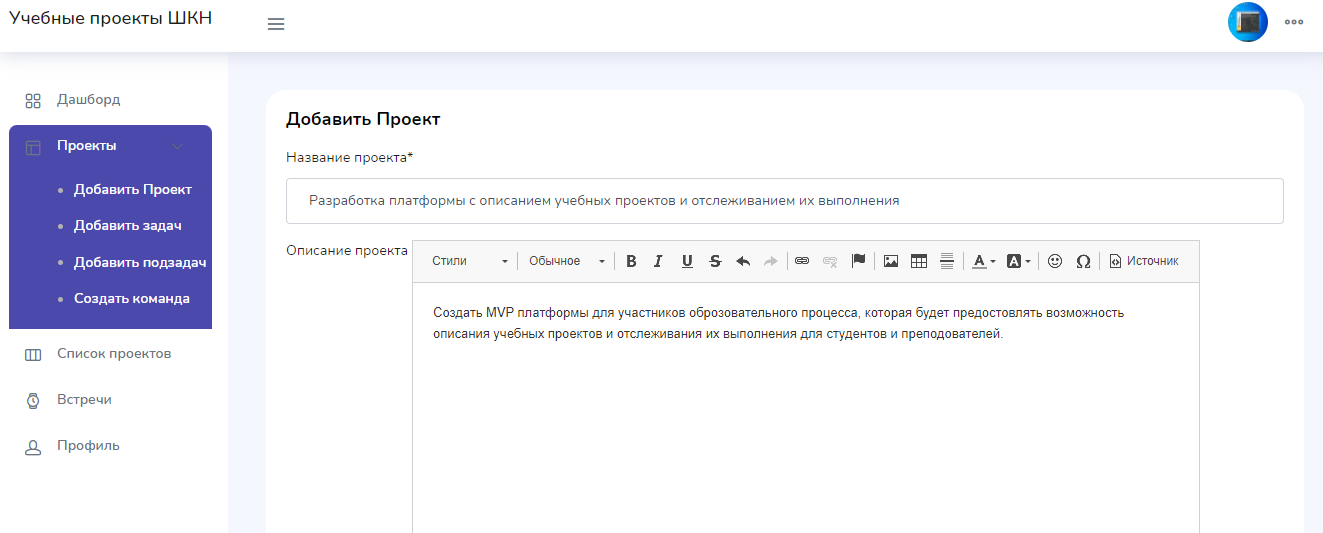


Рис.3.8 Вкладка проекты

При добавлении нового проекта открывается форма, которую нужно заполнить создателю проекта (рис. 3.9). При создании нового проекта преподаватель должен заполнить форму, в которой нужно указать название проекта, описание проекта, используемые технологии для проекта, дата начала проекта, а также дата завершения проекта, после чего можно добавить файл проекта. Для создания проекта необходимо нажать на кнопку создать, но в случае каких-либо неполадок можно отменить создание нажав на кнопку отмены.



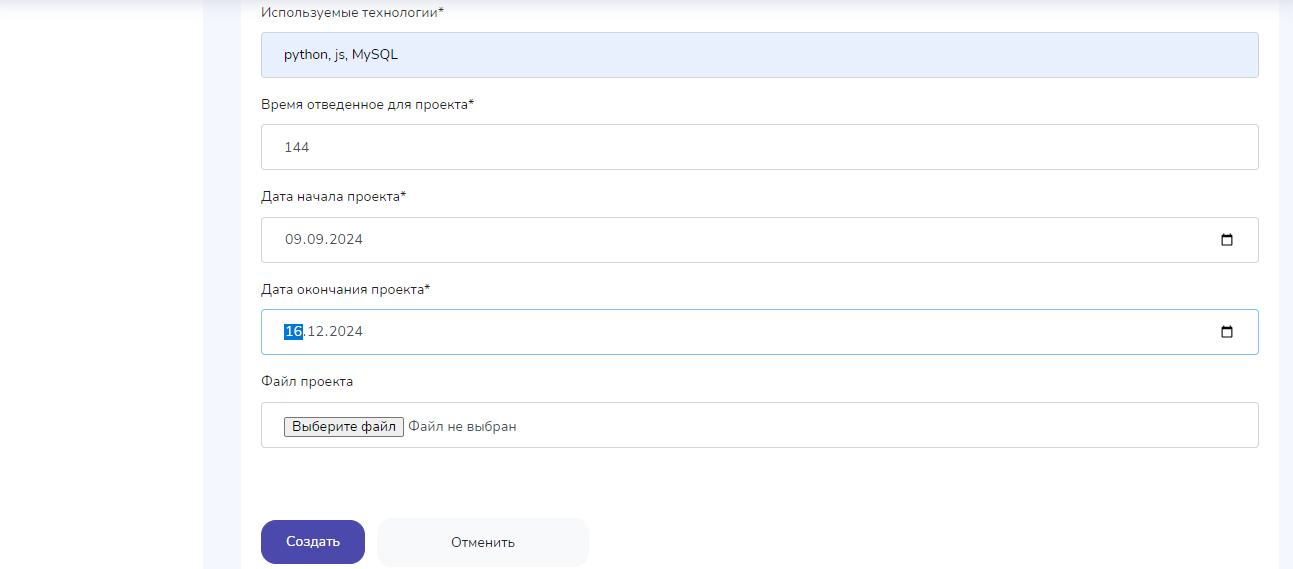


Рис. 3.9 Создание проекта

На этой вкладке можно добавить команду, указав такие параметры как название команды, а также для создания команды преподаватель выбирает студентов команды (рис. 3.10).

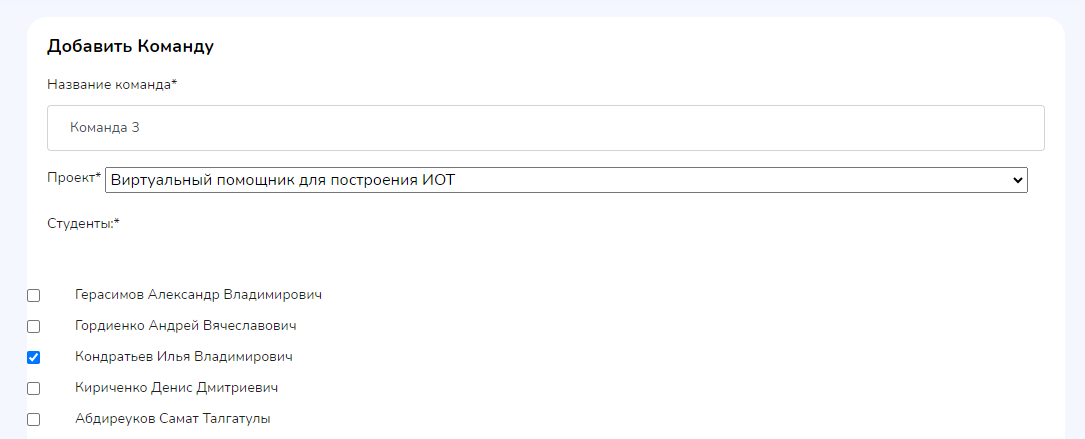


Рис. 3.10 Добавить команду

Дашборд студента включает в себя такие функции как количество доступных ему проектов, количество задач которые он должен выполнить, количество подзадач, а также общее время выделенное для выполнения проектов и задач (рис. 3.11).

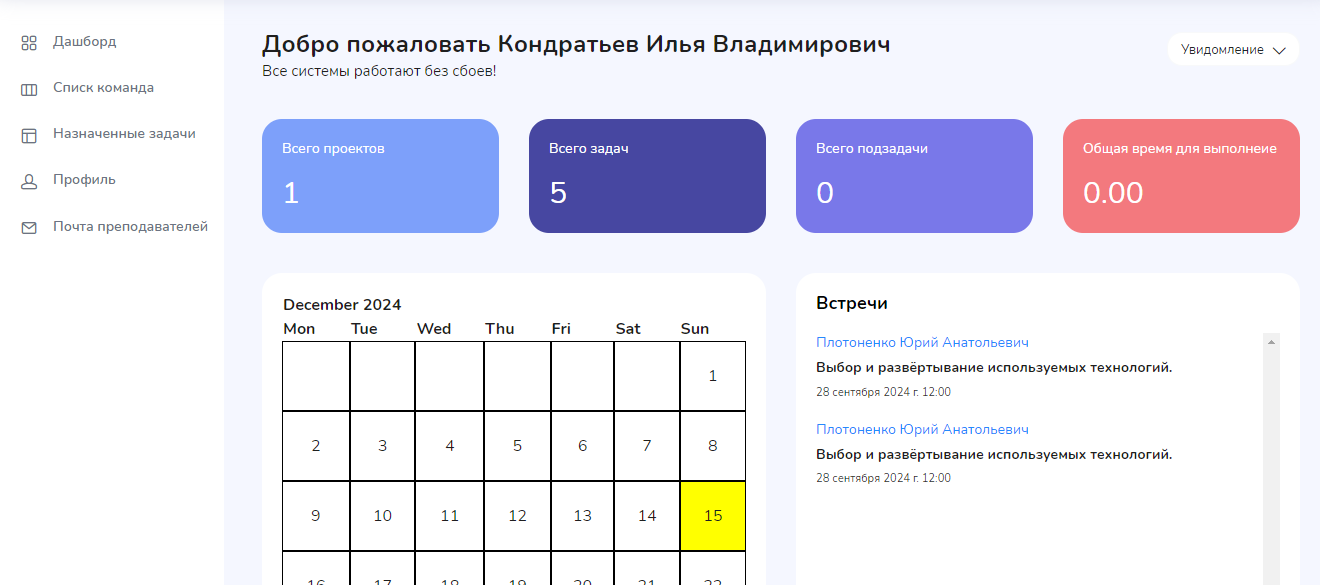


Рис. 3.11 Дашборд студента

В Канбане студента отображены доступные ему задачи (рис.3.12).

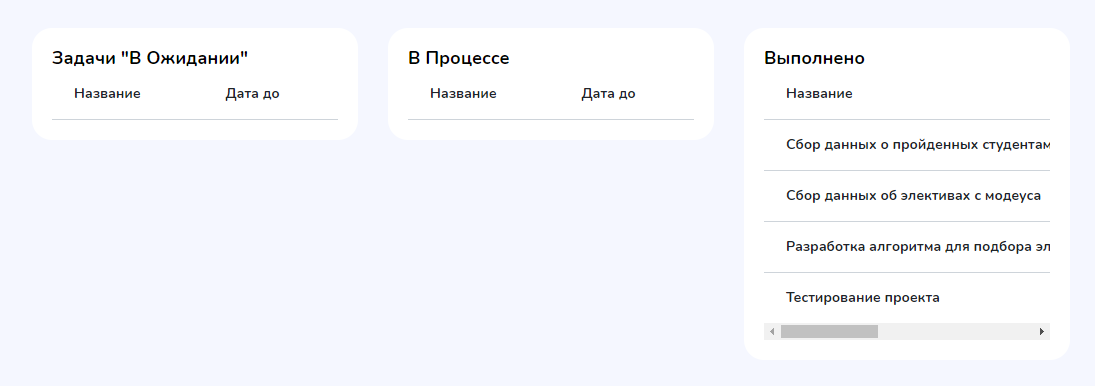


Рис. 3.12 Канбан доска студента

Нажимая список проектов можно и выбирая действия посмотреть задачи отображаются задачи присущие данному проекту (рис. 3.13). В этой страничке также просмотреть диаграмму ганта, где участники указаны разными цветами выполняющие присущие им задачи.

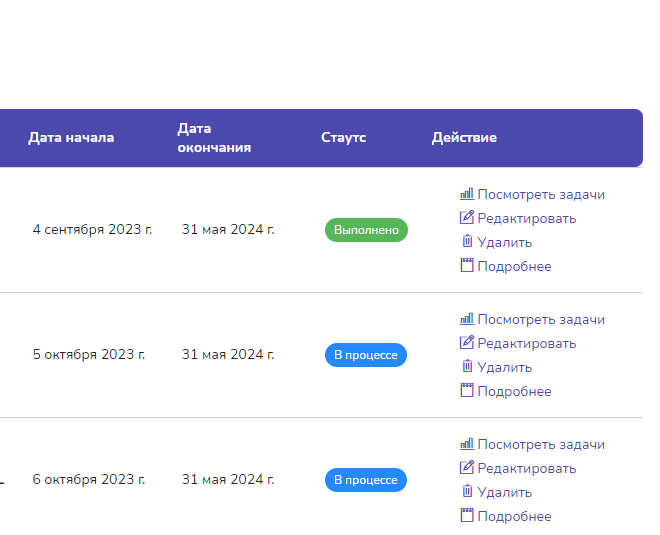
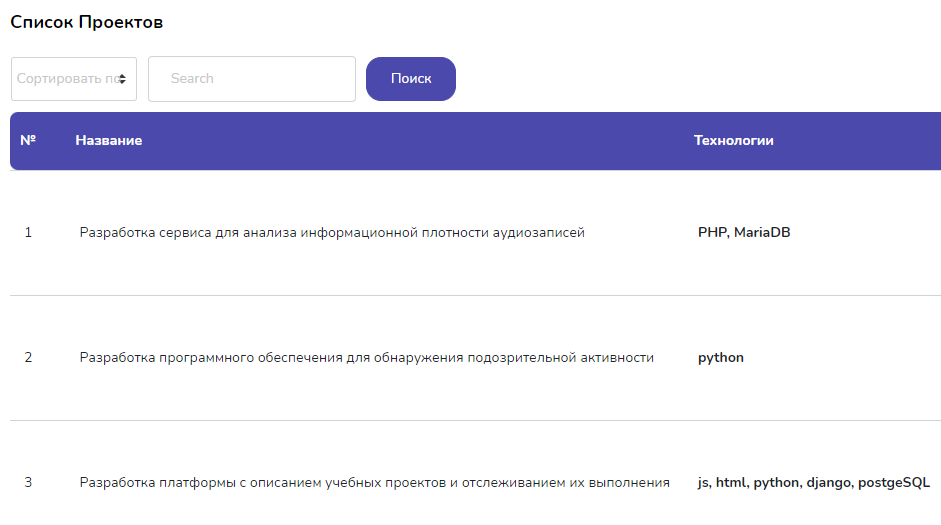


Рис. 3.13 Список проектов

В разделе список проектов указаны такие данные как номер проекта, название проекта, технология, используемая для проекта, дата начала проекта, дата окончание, статус проекта и действие (рис. 3.14). Кроме того, здесь также доступны такие функции как сортировка и поиск проектов.

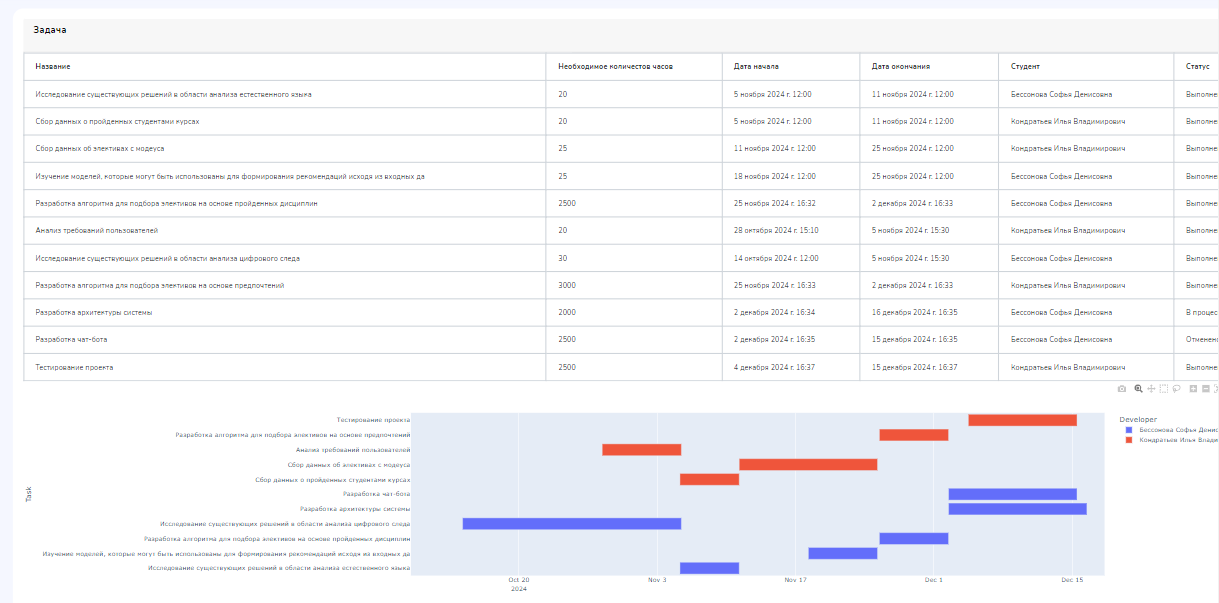


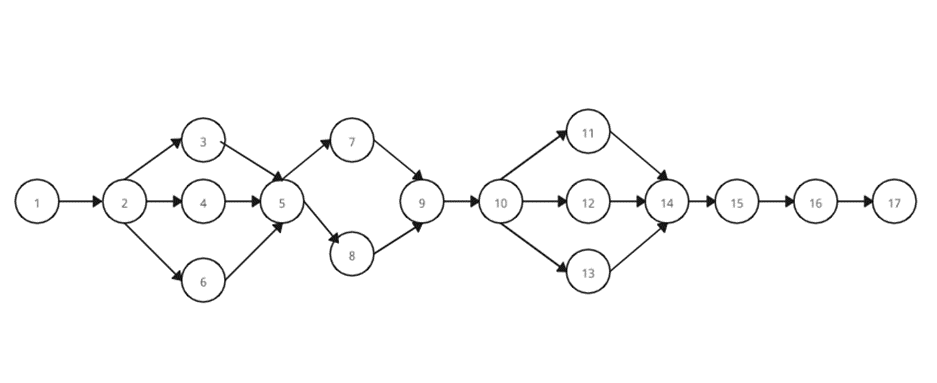
Рис. 3.14 Диаграмма Ганта

## 3.3 ПЛАН РАБОТ

Определен следующий план работ:

|  |  |  |
| --- | --- | --- |
| **№** | **Задача** | **Дедлайн** |
| 1 | Изучить предметную область | 7.09.2024-10.09.2024 |
| 2 | Создать базу данных для хранения информации. | 10.09.2024-15.09.2024 |
| 3 | Проектирование базы данных для хранения информации. | 15.09.2024-23.09.2024 |
| 4 | Подключение БД к серверу. | 23.09.2024-29.09.2024 |
| 5 | Поиск модели для визуализации ведения и выполнения проекта. | 29.09.2024-2.10.2024 |
| 6 | Реализация архитектура бд | 2.10.2024-10.10.2024 |
| 7 | Реализовать канбан доска | 10.10.2024-14.10.2024 |
| 8 | Реализовать диаграмма ганта | 14.10.2024-20.10.2024 |
| 9 | Разработка таймера | 20.10.2024-24.10.2024 |
| 10 | Создание прототипа пользовательского интерфейса. | 24.10.2024-28.10.2024 |
| 11 | Разработка интерфейса преподавателей | 28.10.2024-2.11.2024 |
| 12 | Разработка интерфейса для пользователя. | 2.11.2024-8.11.2024 |
| 13 | Реализовать визуализацию прогресса участников платформы по проектам, задачам и подзадачам для отслеживания их выполнения. | 8.11.2024-15.11.2024 |
| 14 | Разработка функционала для разных ролей пользователей. | 15.11.2024-21.11.2024 |
| 15 | Разработать алгоритм для проверка отчета | 21.11.2024-29.11.2024 |
| 16 | Тестирование | 29.11.2024-4.12.2024 |
| 17 | Написание отчета | 4.12.2024-16.12.2024 |

Задачи в виде сетевого графа:



# ЗАКЛЮЧЕНИЕ

В ходе разработки MVP платформы для отслеживания учебных проектов были решены основные задачи, создана функциональная платформа, которая позволяет пользователям описывать свои учебные проекты, отслеживать их выполнение и делиться результатами с другими участниками.

В дальнейшем планируется продолжить развитие платформы, улучшая её функционал, эффективность и производительность:

* Разработать алгоритмов для проверки отчетов.
* Усовершенствовать систему отслеживания прогресса проектов: добавить статистику по выполнению учебных проектов, эффективности студентов.

Данная платформа может стать хорошим инструментом для студентов и преподавателей, помогая им улучшать и отслеживать учебные проекты и задачи.

# СПИСОК ЛИТЕРАТУРЫ

1. Ankita Paradkar. Task Tracker System - Web Application / Ankita Paradkar, Diksha Bhonde, Shweta Bhond, Priya Rughwani, Shital Chawade // IJSRD. Том. 5, Выпуск 01, 2017 – С. 1-3.
2. Матвеев А.А. Модели и методы управления портфелями проектов. / Матвеев А.А., Новиков Д.А., Цветков А.В.// М.: ПМСОФТ, 2005. – 150-193 с.
3. K. Sandhya, Shwetha Shri. Employee task tracking system // International Research Journal of Modernization in Engineering Technology and Science, Том. 05, Выпуск 07, 2023 – С.
4. Дрейзис Ю.И., Калинина М.В., Зиновьев П.С. The Comparative Analy-sis of the Market of Information Systems for Automation of Activity of Uni-versity. // European Journal of Computer Science, №1, Vol. (2), 2016, pp. 26-37.
5. Рыбанов, А.А. Технология повышения эффективности информацион-ной поддержки, мониторинга и контроля за процессом выполнения вы-пускных квалификационных работ / Рыбанов А.А., Макушкина Л.А., Фа-деева М.В. // Актуальные вопросы профессионального образования. - 2013. - № 10 (т. 10). - C. 122-125.
6. Виштак О.В., Штырова И.А. Автоматизация мониторинга качества в системе дополнительного профессионального образования // Научное обозрение. Педагогические науки. – 2017. – № 5. – С. 14-17;  
   URL: <https://science-pedagogy.ru/ru/article/view?id=1661>
7. Pinto, J K and Slevin, D P (1988) Critical success factors across the project life cycle. Project Management Journal, 67-74.