Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационных компьютерных систем

Дисциплина "Объектно-ориентированное программирование"

К защите допустить:
Руководитель курсовой работы
старший преподаватель
кафедры
_____ А.В.Михалькевич
15.07.2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ЗАГРУЗКИ И ПРОСМОТРА ВИДЕО

БГУИР КР 1-40 05 01-10 № 189 ПЗ

Студент (подпись студента)

Курсовая работа
представлена на проверку
15.07.2025
_____
(подпись студента)

2025

Реферат

БГУИР КР 1-40 05 01-10 № 189 ПЗ, гр. 814302

, РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ЗАГРУЗКИ И ПРОСМОТРА ВИДЕО, Минск: БГУИР - 2025.

Пояснительная записка 29563 с., 5 рис., 0 табл.

Ключевые слова: видеопроигрыватель, веб-приложение

Предмет Объектно-ориентированное программирование, А.В.Михалькевич

Предмет: создание веб-приложения Объект: шаблоны проектирования, разработка пользовательского интерфейса. Цель: разработка веб-приложения для загрузки и просмотра видео с использованием архитектурного шаблона проектирования MVC. Методология проведения работы: в процессе решения поставленных задач спроектирован и разработан простой и удобный интерфейс веб-приложения, разработана логика приложения с использованием паттерна MVC, изучена и применена работа с базой данных. Результаты работы:В ходе курсового проекта было создано веб-приложение для загрузки и просмотра видео с использованием современных инструментов разработки сайтов. Область применения результатов: удовлетворение пользователей удобной реализацией видео файлов и с его подробным описанием, так же загрузкой видео. Ссылка на онлайн-репозиторий GitHub:

-Subject: creating a web application Object: design templates, user interface development. Goal: develop a web application for downloading and viewing videos using the MVC architectural design template. Methodology of work: in the process of solving the tasks, a simple and convenient web application interface was designed and developed, the application logic using the MVC pattern was developed, and work with the database was studied and applied. Result of work:During the course of the course project, a web application was created for downloading and viewing videos using modern site development tools. Scope of results: satisfaction of users with convenient implementation of video files and with its detailed description, as well as video loading. Link to the github online repository:

Содержание

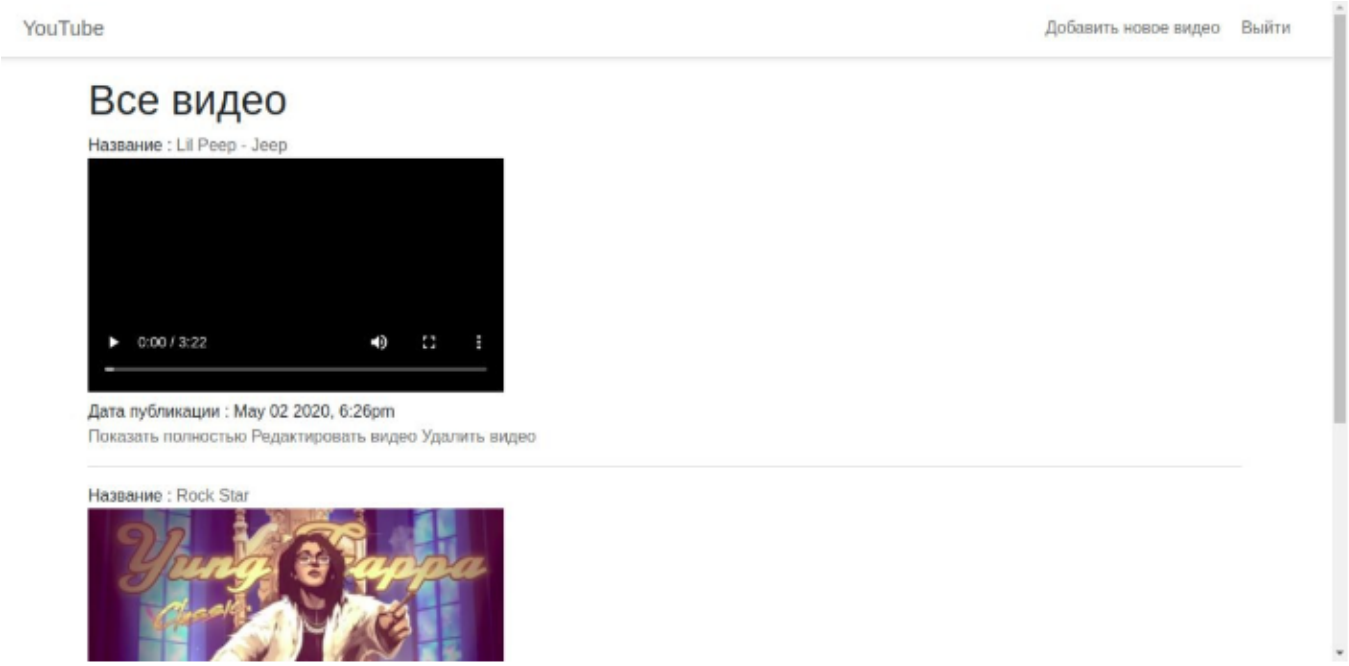
Введение
1 описание проекта
2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ
3 ИНСТРУМЕНТАРИЙ
4 АРХИТЕКТУРНЫЙ ШАБЛОН ПРОЕКТИРОВАНИЯ MVC
5 РАЗРАБОТКА ПРОГРАММЫ
Заключение
Список использованных источников
Приложения

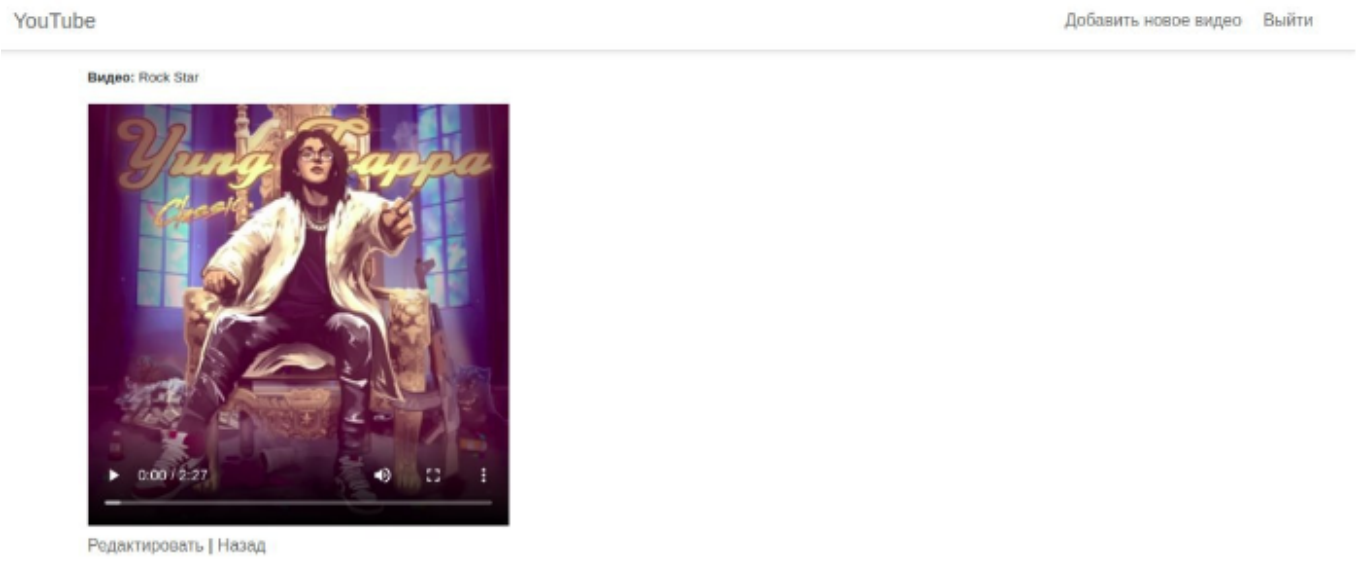
Введение

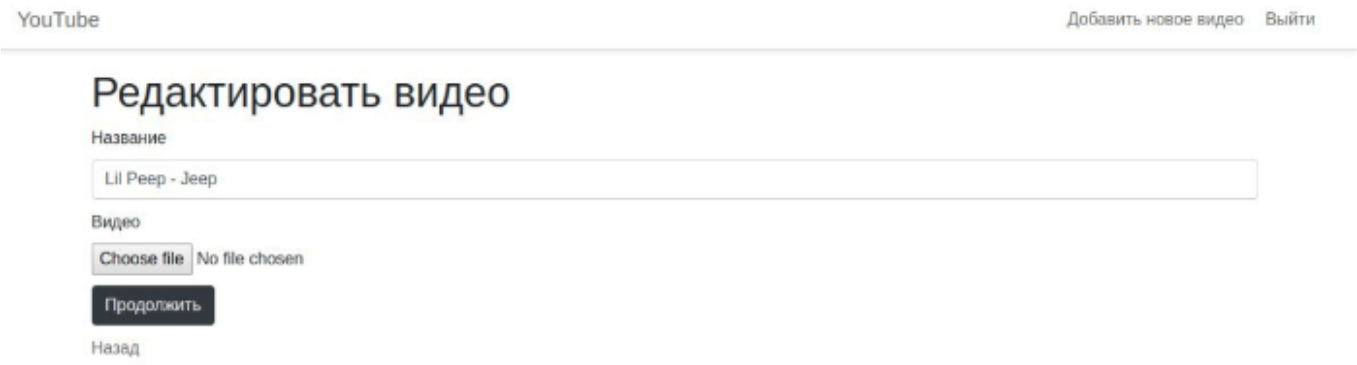
В наше время интернет развивается с огромной скоростью и проникает во все сферы жизни. Так же в сферу кино и просмотра видео. Интернет стал достаточно удобной и дешевой площадкой для видеохостинга, огромное количество людей размещают свои видео в сети, а так же смотрят видео и фильмы в режиме on-line. Люди могут проводить свой досуг не выходя из дома благодоря совершенствованию интернет площадок с видео файлами, люди могут смотреть фильмы не посещая кинотеатры. Так же современные интернет площадки с видео файлами предоставляют людям партнерские программы, с помощью которых люди могут зарабатывать деньги. Большенство людей использующие сеть интерент регулярно работают из офисов и проводящие много времени за компьютером, по-этому возможность просмотра чего-либо их интересующего, не покидая дом или рабочее место, является огромным плюсом. Цель данного курсового проекта: разработка веб-приложения для загрузки и просмотра видео. Задачи: изучение современных инструментов разработки сайтов, реализация веб-сайта, создание баз данных для хранения информации.

1 описание проекта

Курсовой проект представляет собой веб-сайт для загрузки и просмотра видео, с его подробным описанием, а также информацией о самих видео, разработанный при помощи Ruby фреймворка Ruby on Rails, с использованием языков разметки HTML и CSS. Вся изменяющаяся информация на сайте хранится при помощи баз данных. Поддерживается возможность загрузки и редактирования видео.







2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Для разработки веб-сайта был выбран язык Ruby. Ruby - динамический, рефлективный, интерпретируемый высокоуровневый объектно ориентированный язык программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, «сборщиком мусора» и многими другими возможностями.

Ruby реализует идеологию «всё — объект», то есть любая единица данных является объектом — экземпляром некоторого класса, к которому применимы все синтаксические средства, предназначенные для работы с объектами. В этом смысле язык не содержит встроенных примитивных типов данных. Условно таковыми можно считать типы,

предоставляемые интерпретатором и системной библиотекой, используемые наиболее часто и не требующие для использования специального указания имени класса.

Ruby - полностью объектно-ориентированный язык. В нём все данные являются объектами, в отличие от многих других языков, где существуют примитивные типы. Каждая функция - метод.

В частности, использовался фреймворк Ruby on Rails – это фреймворк, написанный на Ruby, реализует архитектурный шаблон Model-View-Controller для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером баз данных.

Важным преимуществом Ruby on Rails является то, что его используют многие компании, в особенности стартапы, по ряду причин. Во-первых, сам язык Ruby настолько ясен, прост и логичен, что его часто советуют новичкам.

Во-вторых, быстрая скорость создания приложений. Процесс, включающий задумку создания приложения до его онлайн-реализации, может занять всего лишь несколько часов, чем могут похвастаться далеко не многие фрэймворки.

Ruby on Rails — полное название для Rails. Связано это с тем, что большинство программистов просто не употребляют первую часть, но все прекрасно понимают, что она означает. Гипотетически вместо Rails можно использовать другие фрэймворки, как, например, Sinatra. Но в настоящее время 99% программистов Ruby осуществляют веб-разработку на Rails, поэтому выбор других фрэймфорков будет просто неэффективен.

Ruby — регистро-зависимый язык, прописные и строчные буквы в идентификаторах являются различными. Все ключевые слова языка, за двумя исключениями, пишутся в нижнем регистре.

Основные преимущества Ruby on Rails:

Процесс программирования достаточно быстрый;
Rails имеет отличные встроенные решения в области тестирования;;
Имеет лаконичный и простой синтаксис;

3 ИНСТРУМЕНТАРИЙ

SQLite 3 - компактная встраиваемая СУБД.

Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном

файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала.

Преимущества SQLite 3:

Файловая: вся база данных хранится в одном файле, что облегчает перемещение.

Стандартизированная: SQLite использует SQL; некоторые функции опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), однако, есть и некоторые новые.

Отлично подходит для разработки и даже тестирования: во время этапа разработки большинству требуется масштабируемое решение. SQLite, со своим богатым набором функций, может предоставить более чем достаточный функционал, при этом будучи достаточно простой для работы с одним файлом и связанной сишной библиотекой.

Каждое значение в любом поле любой записи может быть любого из этих типов, независимо от типа, указанного при объявлении полей таблицы. Указанный при объявлении поля тип хранится для справки в его исходном написании, и используется в качестве основы для выбора предпочтений (так называемое «type affinity»: это подход, редко встречающийся в других СУБД) при выполнении неявных преобразований типов на основании похожести этого названия типа на что-либо, знакомое SQLite. В этот алгоритм зашит обширный перечень практикуемых в других СУБД вариантов названий типов данных. Если безопасного преобразования записываемого значения в предпочитаемый тип не получается, SQLite записывает значение в его исходном виде.

RubyMine

RubyMine-коммерческая IDE для разработки программного обеспечения на Ruby компании JetBrains.

RubyMine создан на основе IntelliJ IDEA того же производителя. Поддерживает популярные библиотеки, используемые в Ruby-приложениях (в том числе Bundler, RSpec, Shoulda, Cucumber, Git)

Основные возможности:

Редактор кода с подсветкой синтаксиса, функцией автодополнения
Анализ кода на лету с возможностью моментального исправления
Быстрая навигация по элементам проекта и элементам кода
Полноценная поддержка языков HTML/HAML, CSS/Sass/Less, JavaScript/CoffeeScript
История изменений с возможностью сравнения версий, как в рамках поддерживаемых VCS, так и в рамках локальной истории IDE

Светлая и темная цветовые схемы с возможностью индивидуальной настройки
Пошаговый отладчик Ruby-кода с возможностью выставления точек остановки и трассировки
Встроенные схемы клавиш быстрого доступа на базе схем Textmate, NetBeans, Eclipse, Emacs, Vi/Vim

GitHub

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы. Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. Первый частный репозиторий был создан 12 января 2008. К концу 2011 года в проекте уже было зарегистрировано более миллиона пользователей и более двух миллионов репозиториев. По состоянию на март 2017 года на сайте существовало более 58 миллионов репозиториев.

Использование системы контроля GIT

Система предполагает защищенное хранение данных, т.е. любой хранимый в ней блок имеет множество клонов. При повреждении какого-либо файла есть возможность своевременно заменить его копией. Для уменьшения объема данных проекта часто используется дельта-компрессия – такой вид хранения, при котором хранятся не сами версии файла, а только изменения между последовательными ревизиями.

Устанавливается Git на Windows очень просто – при помощи инсталлятора. После установки есть возможность пользоваться SSH клиентом и стандартной графической версией. Далее создаётся локальный репозиторий. Для этого нужно зайти в папку с проектом и прописать в командной строке.

Команды для корректной реализации работы:

*«git add *»* - добавляет все новые файлы в область подготовленных файлов;

«git commit –m “commit comment here”» - фиксирует изменения;

«git push» - отправляет файлы в репозиторий;

«git status» - показывает текущий статус репозитория;

«git pull» - скачивает файлы из удаленного репозитория в локальный

4 АРХИТЕКТУРНЫЙ ШАБЛОН ПРОЕКТИРОВАНИЯ MVC

Если в проектировании системы паттерны практических решений могут быть и не использованы, то использование архитектурных шаблонов в проектировании любой системы — обязательно. Понимание и использование архитектурного шаблона, применяемого в проектировании системы обязует программиста писать хороший код.

В группе архитектурных шаблонов выделяют следующие.

Простой шаблон
MVC
HMVC
MVVM

Архитектура приложения – это логическая структура, описывающая отдельные компоненты, их свойства и связи в виде единой системы.

Паттерны – это описания схем детализации отдельных подсистем приложения и взаимосвязей между ними

MVC - это аббревиатура, которая расшифровывается так: Model-View-Controller.

MVC - это именно набор архитектурных идей и принципов для построения сложных систем с пользовательским интерфейсом.

Модель (Model) представляет собой объектную модель некой предметной области, включает в себя данные и методы работы с этими данными, реагирует на запросы из контроллера, возвращая данные и изменяя своё состояние, при этом модель не содержит в себе информации, как данные можно визуализировать.

Представление (View) отвечает за отображение информации (визуализацию), одни и те же данные могут представляться различными способами, например, коллекцию объектов можно представить как в табличном виде, так и списком.

Представление также перехватывает действие пользователя, которое затем передается Контроллеру. Характерным примером этого является кнопка, генерируемая Представлением. Когда пользователь нажимает ее, запускается действие в Контроллере.

Кроме этого определение Представления как файла шаблона также является неточным. Но это не вина одного человека, а результат множества ошибок различных разработчиков, которые приводят общему заблуждению. После чего они неправильно объясняют это другим. На самом деле Представление - это намного больше, чем просто шаблон. Но современные MVC-ориентированные фреймворки до такой степени впитали этот подход, что никто уже не заботится о том, поддерживается ли верная структура MVC или нет.

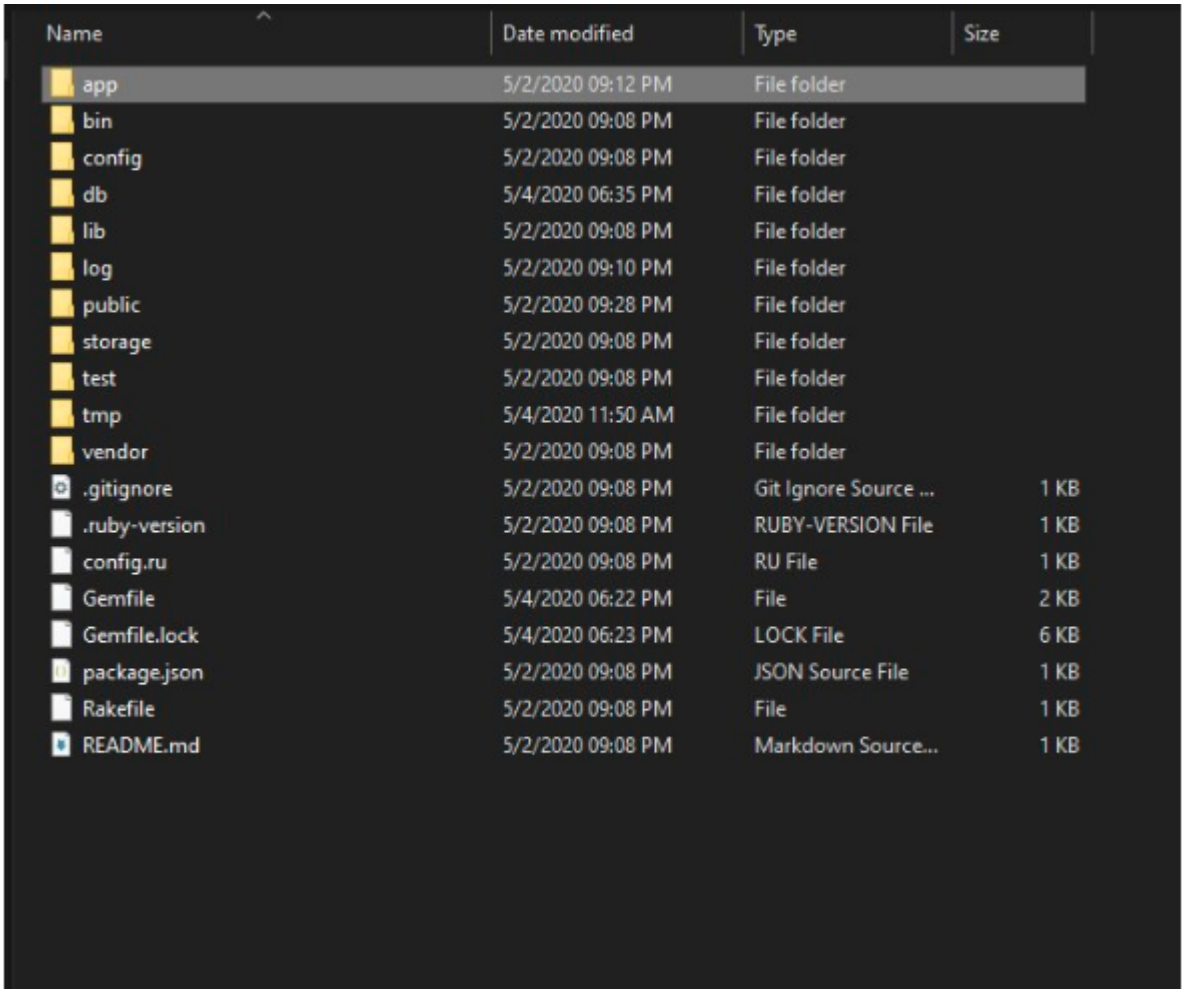
Контроллер (Controller) обеспечивает связь между пользователем и системой, использует модель и представление для реализации необходимой реакции на действия пользователя, как правило, на уровне контроллера осуществляется фильтрация полученных данных и авторизация (проверяются права пользователя на выполнение действий или получение информации).

Контроллер получает задачи на выполнение только когда пользователь взаимодействует с Представлением, и каждая функция зависит от взаимодействия пользователя с Представлением. Наиболее распространенная ошибка разработчиков заключается в том, что

они путают Контроллер со шлюзом, поэтому присваивают ему функции и задачи, которые относятся к представлению.

5 РАЗРАБОТКА ПРОГРАММЫ

Каталог после установки фреймворка и написания кода



Name	Date modified	Type	Size
app	5/2/2020 09:12 PM	File folder	
bin	5/2/2020 09:08 PM	File folder	
config	5/2/2020 09:08 PM	File folder	
db	5/4/2020 06:35 PM	File folder	
lib	5/2/2020 09:08 PM	File folder	
log	5/2/2020 09:10 PM	File folder	
public	5/2/2020 09:28 PM	File folder	
storage	5/2/2020 09:08 PM	File folder	
test	5/2/2020 09:08 PM	File folder	
tmp	5/4/2020 11:50 AM	File folder	
vendor	5/2/2020 09:08 PM	File folder	
.gitignore	5/2/2020 09:08 PM	Git Ignore Source ...	1 KB
.ruby-version	5/2/2020 09:08 PM	RUBY-VERSION File	1 KB
config.ru	5/2/2020 09:08 PM	RU File	1 KB
Gemfile	5/4/2020 06:22 PM	File	2 KB
Gemfile.lock	5/4/2020 06:23 PM	LOCK File	6 KB
package.json	5/2/2020 09:08 PM	JSON Source File	1 KB
Rakefile	5/2/2020 09:08 PM	File	1 KB
README.md	5/2/2020 09:08 PM	Markdown Source...	1 KB

Рисунок 4 – Каталог с веб-сайтом

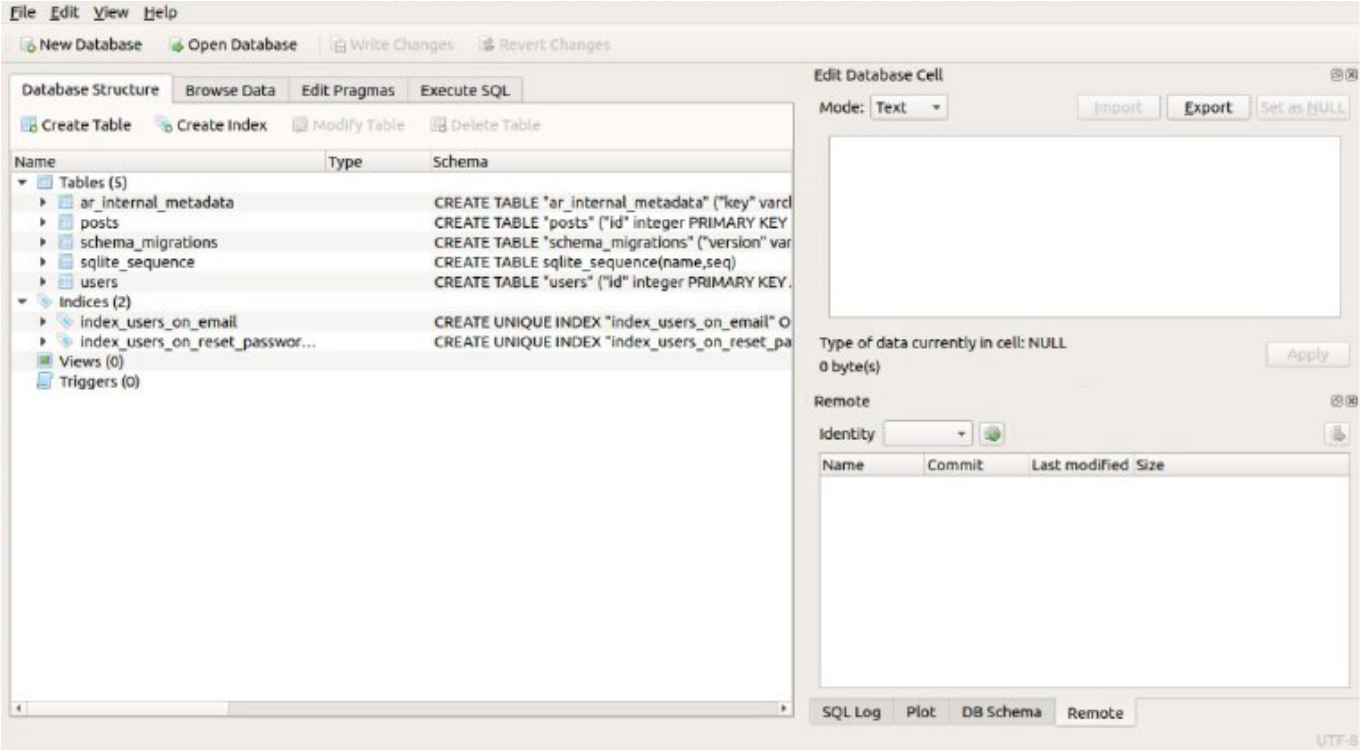


Рисунок 9 – База данных проекта

Заключение

В ходе курсового проекта было создано веб-приложение для загрузки и просмотра видео. Веб-приложение было создано при помощи языка программирования Ruby и Ruby-фреймворка Ruby on Rails, СУБД SQLite3, языка разметки html, языка описания внешнего вида CSS. Также проект был занесён на онлайн-репозиторий GitHub.

Список использованных источников

1. [url] **Статьи о различных технологиях программирования** https://ru.wikipedia.org/
2. [url] **Основы Git** https://habr.com/ru/post/149356/
3. [url] **SQLite 3** https://habr.com/ru/post/149356/
4. [url] **GitHub** https://ru.wikipedia.org/wiki/GitHub
5. [url] **MVC** https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami
6. [url] **Основы Git** : https://git-scm.com/

Приложения