Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Уральский федеральный университет   
имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и технологий

Кафедра «Теплофизика и информатика в металлургии»

**Разработка веб-приложения для расчета теплообмена в противоточном движении кусковых материалов и газов в плотном продуваемом слое**

**ОТЧЕТ**

**по практической работе №2**

**по дисциплине «Моделирование процессов и систем в металлургии»**

Направление 09.03.02 «Информационные системы и технологии» (уровень бакалавриата)

Студент

группы НМТ-303902 А. В. Лысова

Преподаватель:

доцент И. А. Гурин

Екатеринбург

2022

Формулировка задания

Разработать веб-приложение для расчета теплообмена в противоточном движении кусковых материалов и газов в плотном продуваемом слое.

Реализовать приложение на языке программирования C# с помощью технологии ASP.NET MVC Core.

Интерфейс программы должен предусматривать возможность редактирования исходных данных, отображение результатов в табличной форме и графическое представление изменения температуры материала и газа по высоте слоя и разницы между ними. Возможность сохранения вариантов расчета.

Ход работы

Для работы необходимо создать проект на языке С# используя ASP.NET Core (модель-представление- контроллер) – рисунок 1.

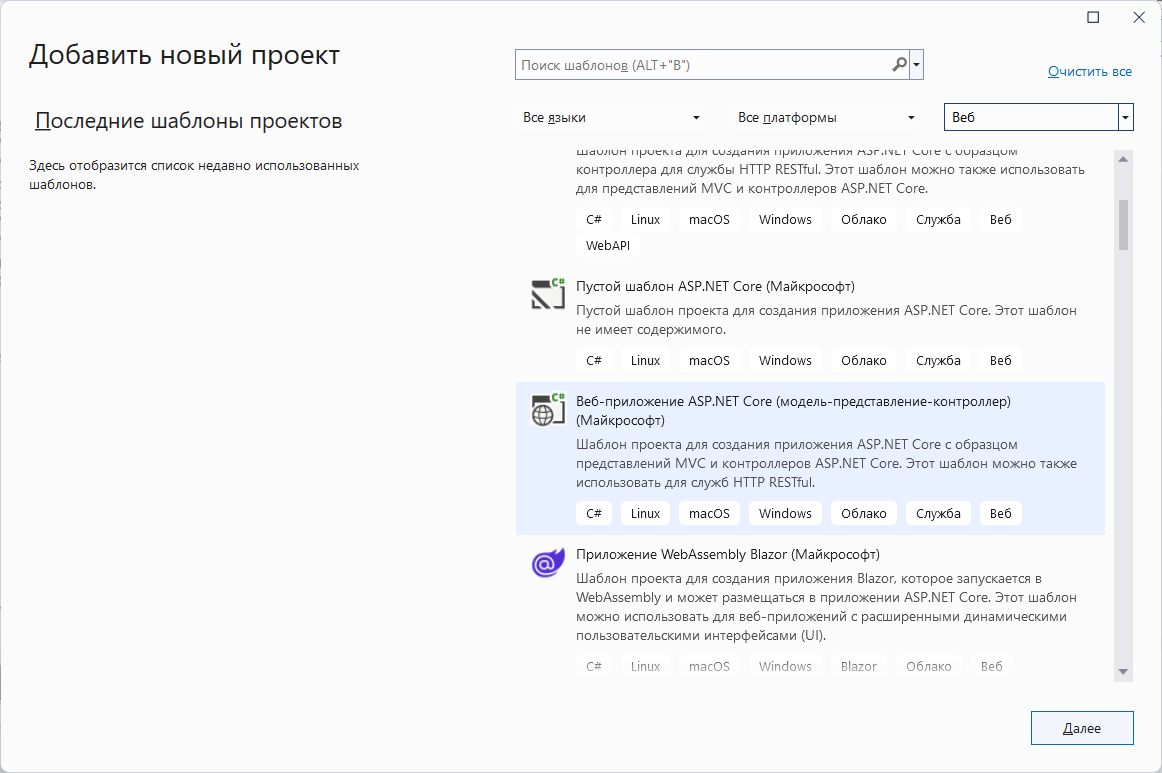


Рис.1 – создание проекта

Далее создаётся библиотека классов, в которой будут храниться классы для получения (рисунок 2), отправления данных (рисунок 3), а также класс с необходимыми расчётами (рисунок 4).

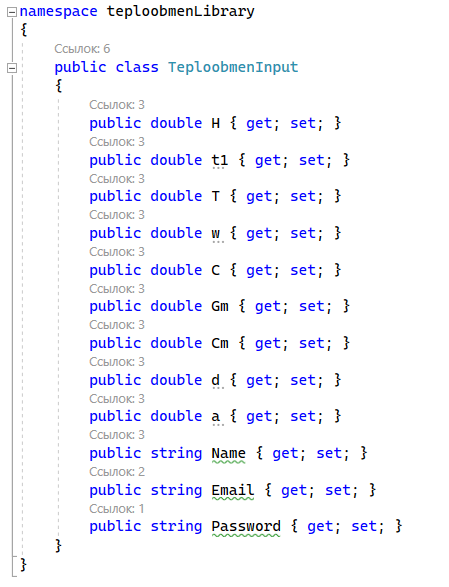


Рис.2 – класс для получения начальных данных

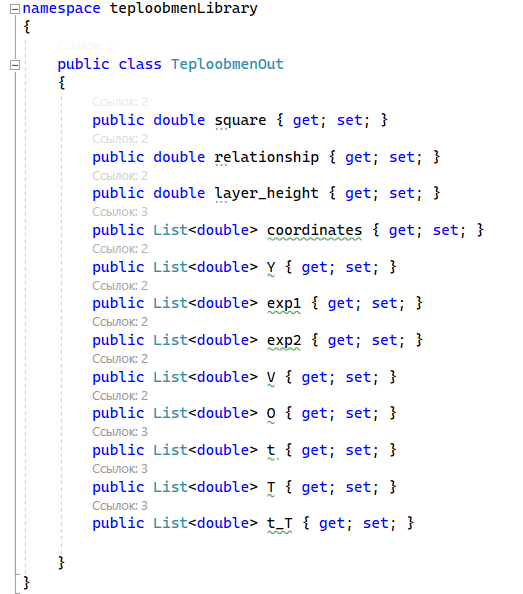


Рис.3 – класс для отправления результатов расчётов

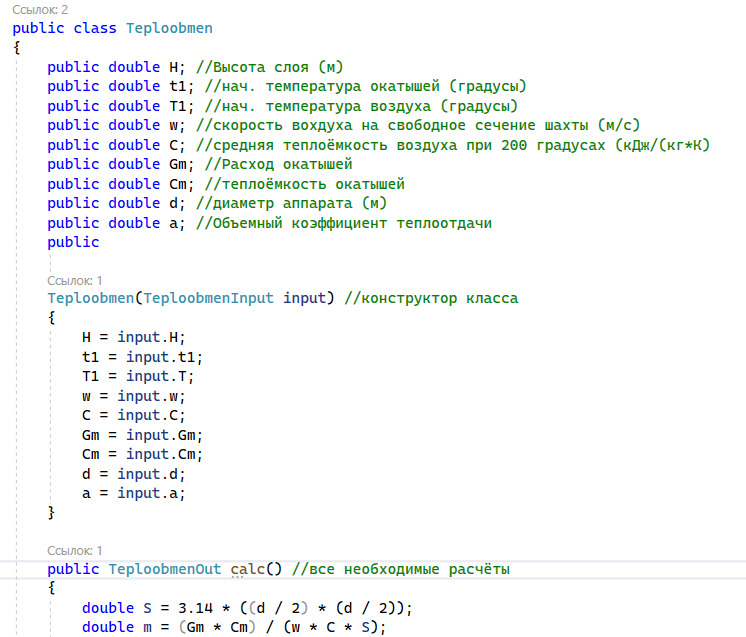


Рис. 4 – класс для расчётов

C помощью диспетчера NuGet пакетов добавляем следующие пакеты.

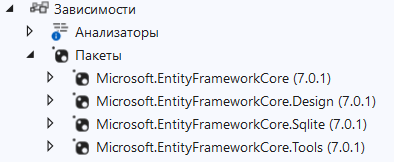


Рис. 5 – дополнительные пакеты

Создаем папку Data, в которую помещаем классы Users, (рисунок 6) Variant (рисунок 7), в которые записываем структуру нашей базы данных и класс ApplicationContext (рисунок 8), который связывает базу данных с приложением.

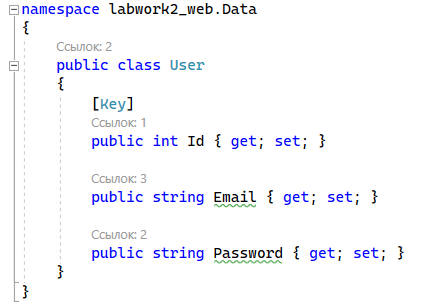


Рис. 6 – класс User

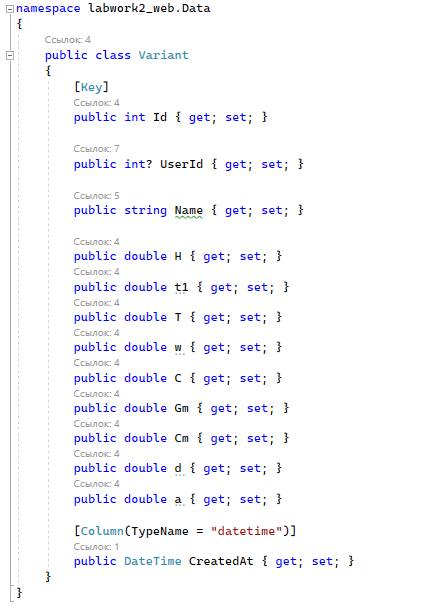


Рис. 7 – класс Variant

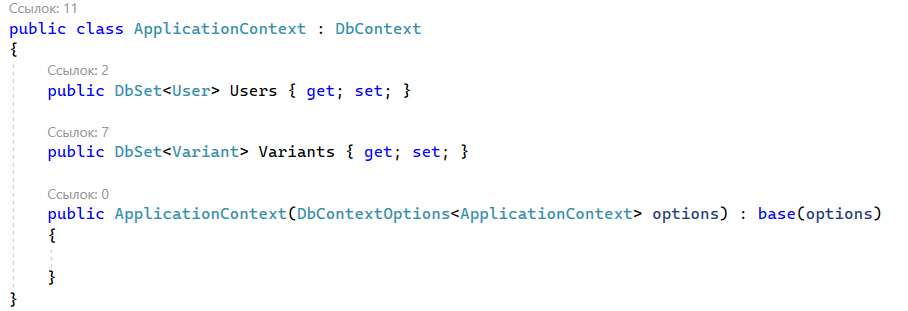


Рис. 8 – класс ApplicationContext

Добавляем в файл appsettings.json строку подключения к базе данных – рисунок 9.



Рис. 9 – файл appsettings.json

В файл program.cs добавляем строку подключения – рисунок 10, а также метод, чтобы программа запоминала вход с устройства и при открытии заходила в последний аккаунт.

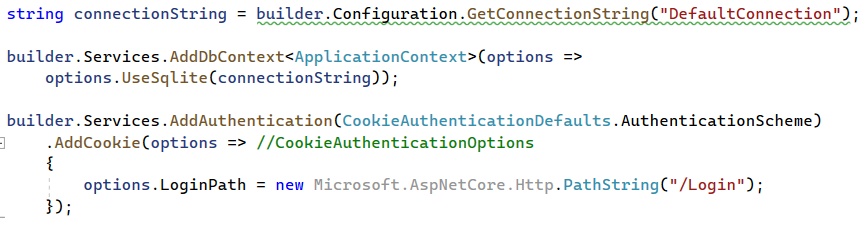


Рис. 10 – подключение к базе данных

С помощью консоли диспетчера пакетом создаем базу данных с помощью команды «add-migration».

Нам понадобятся четыре контроллера, которые мы добавляем в папку Controllers – HomeController, LoginController, RegistrController и UserController.

Добавляем в HomeController два метода: первый выводит на экран доступные варианты данных (для авторизованных и не авторизованных пользователей), а второй при определённых правах позволяет удалить вариант из списка – рисунок 11.

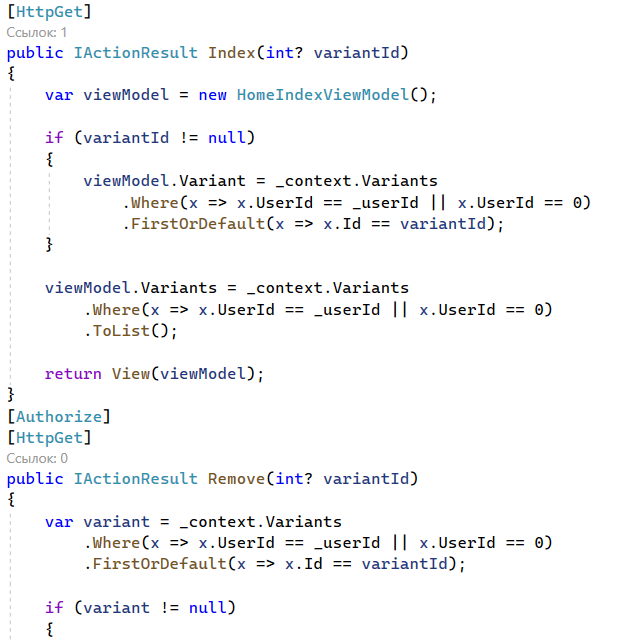


Рис. 11 – методы вывода на экран и удаления вариантов

Далее добавляем метод, который принимает входные параметры, сравнивает название варианта если такого варианта нет добавляет новый вариант в базу данных, а также выводит результат на экран – рисунок 12.

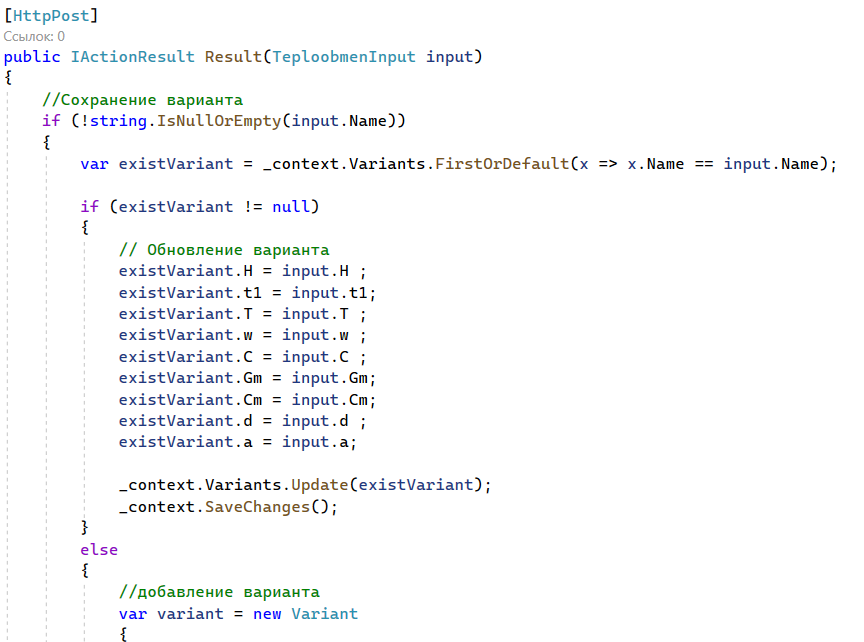


Рис.12 – методы добавления вариантов в базу данных и расчёта

Контроллер LoginController предназначен для авторизации пользователей – рисунок 13.

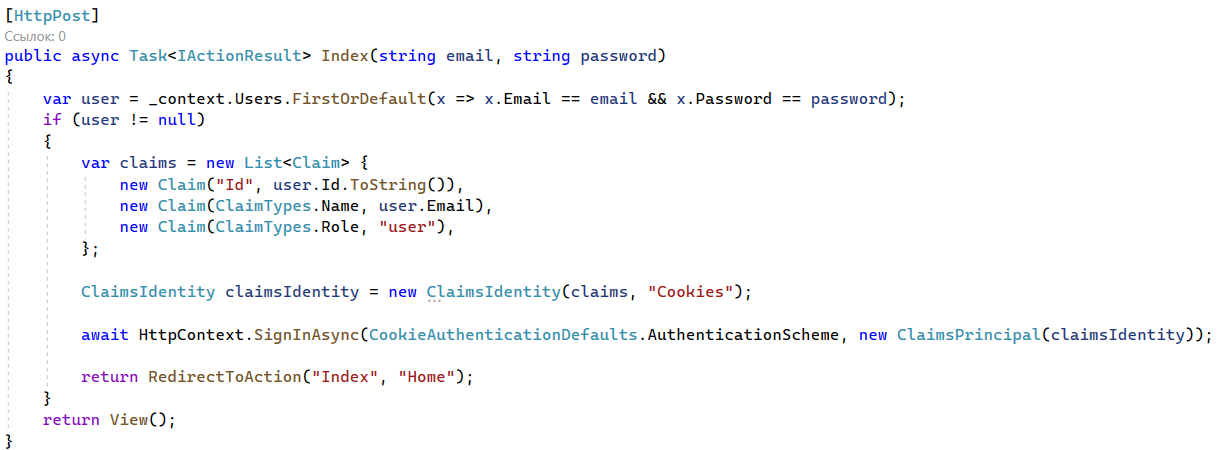


Рис.13 – метод контроллера LoginController для авторизации пользователей

Контроллер RegistrController предназначен для регистрации пользователей, который отправляет информацию в базу данных – рисунок 14

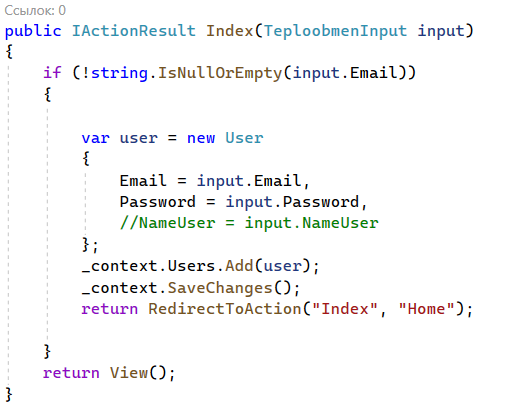


Рис.14 – метод контроллера RegistrController для регистрации пользователей

Контроллер UserController предназначен для проверки авторизации пользователя (если авторизован, то появляется вкладка «личный кабнет») – рисунок 15

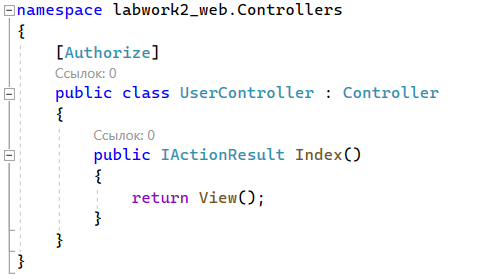


Рис.15 – контроллер UserController

Так же нам необходимо создать пять представлений Views два представления в папке Home – Index и Result, а также по одному представлению Index в папках User, Login, Registr (рисунок 16)

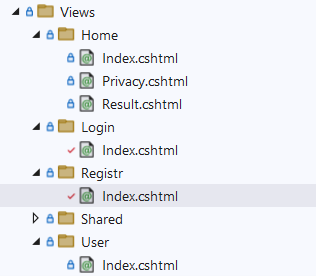


Рис. 16 – структура папки Views

Для страницы расчётов есть две страницы. Index (рисунки 17-1) это страница, которая открывается при запуске программы, там находятся таблица вариантов, а также поля для вписывания исходных данных и кнопка «рассчитать».

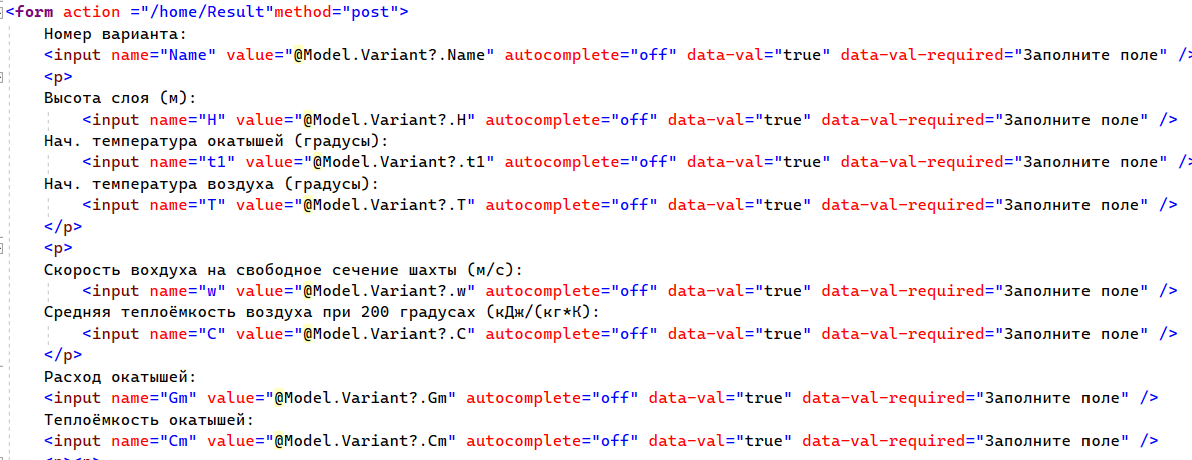


Рис. 17 – ввод исходных данных

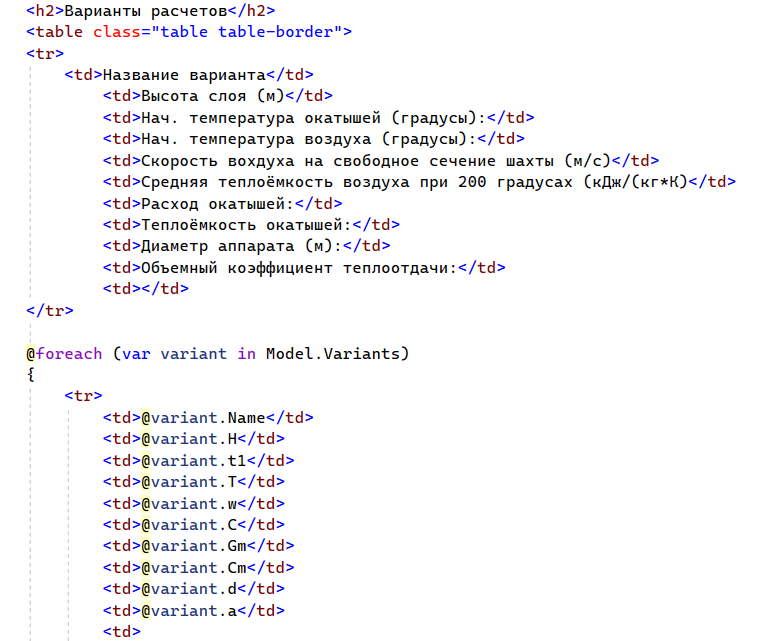


Рис. 18 – представление таблицы вариантов из базы данных

Итоговая главная страница представлена на рисунке 19.

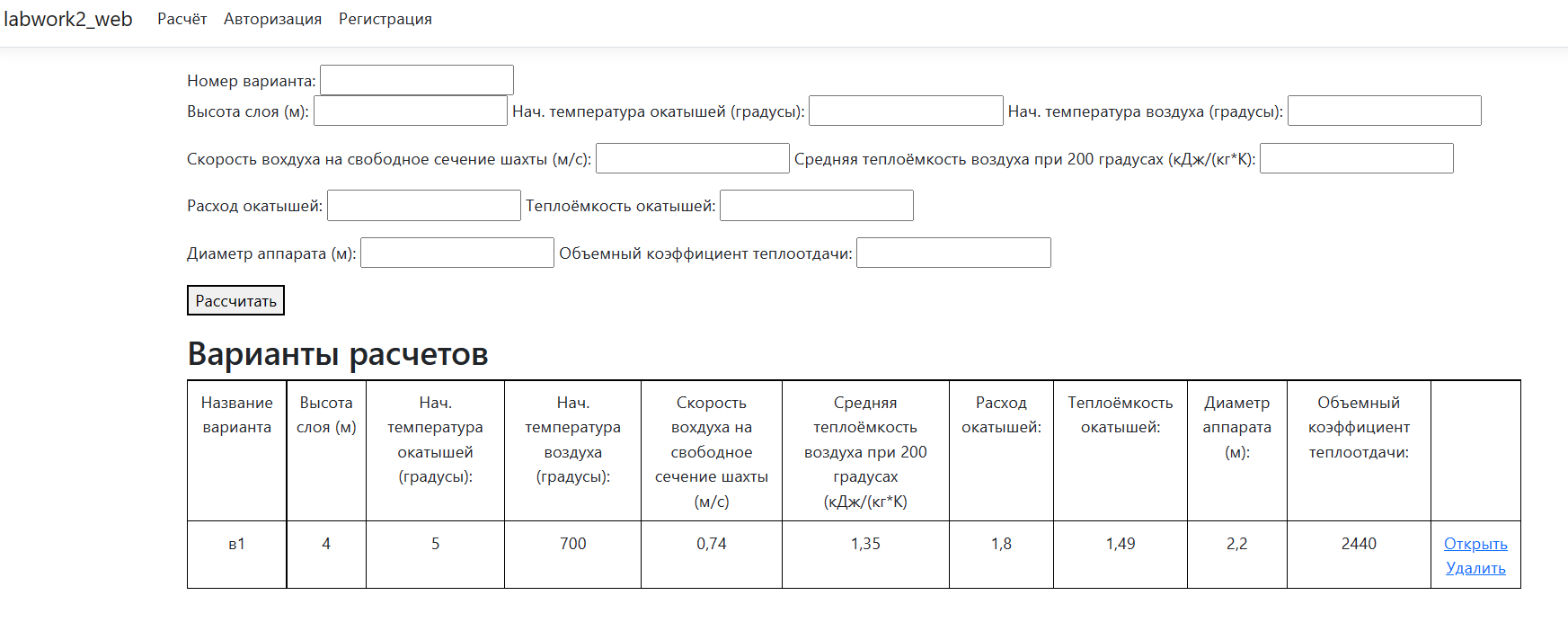


Рис. 19 – вид главной страницы

Вторая страница Result предназначена для представления результатов расчёта и вывода графика – рисунки 20,21.

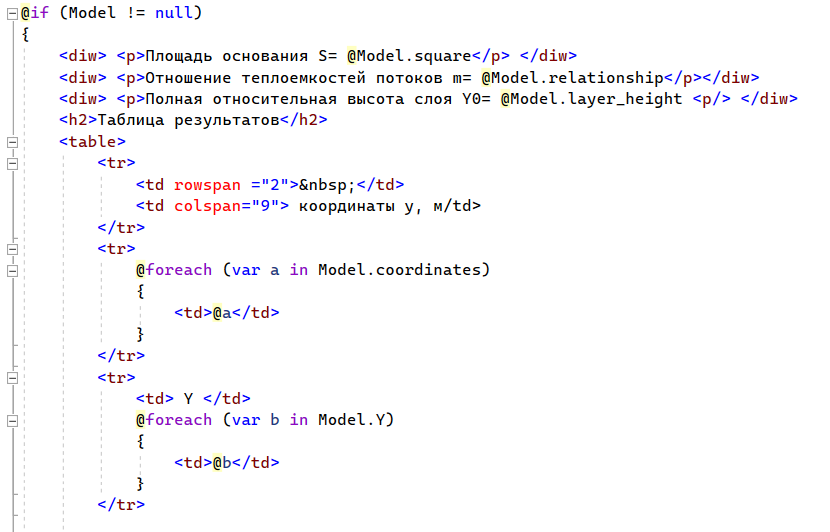


Рис. 20 – вывод таблицы результатов



Рис. 21 – создание графика

После нажатия на кнопку «рассчитать» на главной странице открывается страница с таблицей результатов и графиком – рисунок 22.

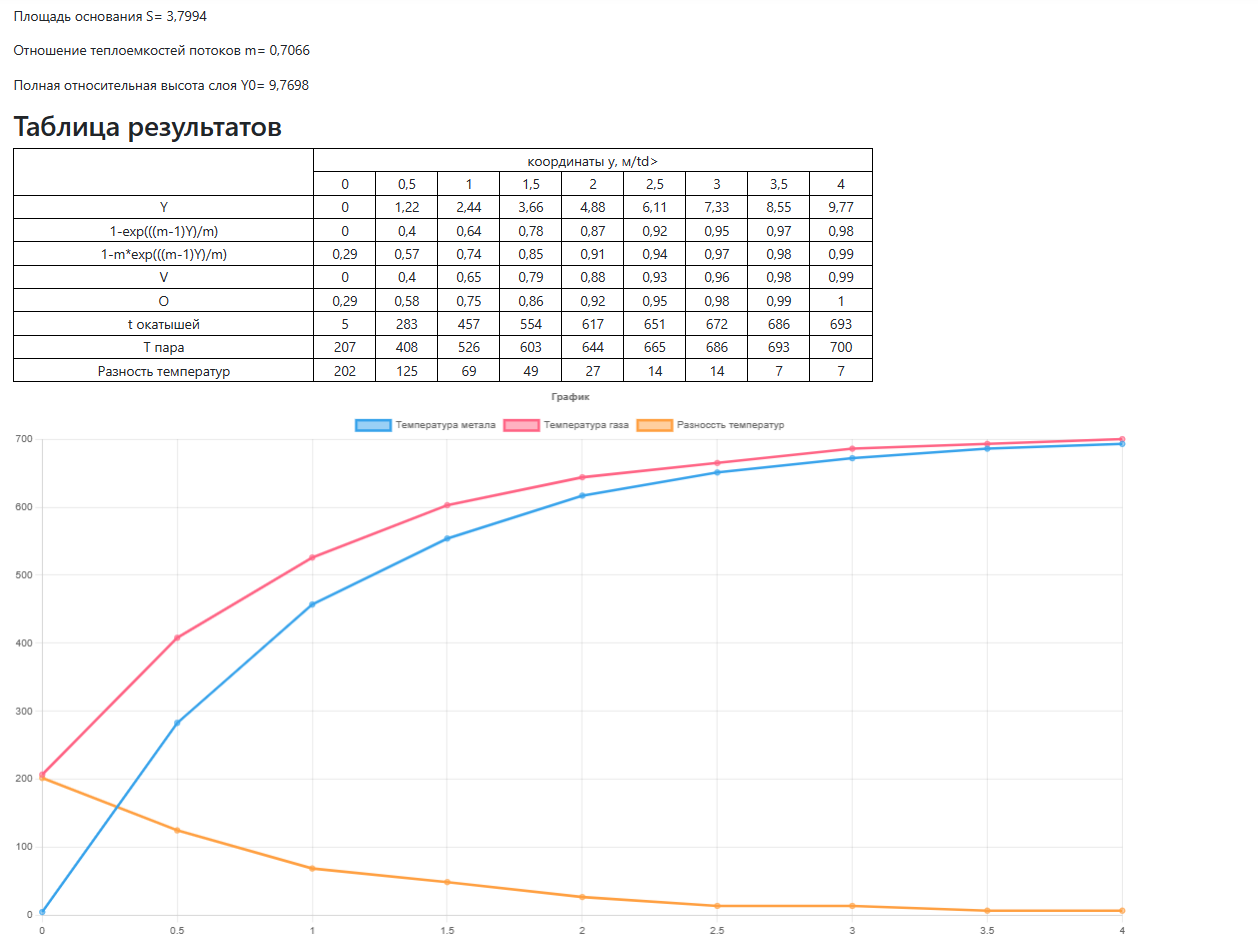


Рис. 22 – страница результатов

В папке Login содержится файл Index, который представляет собой страницу «Авторизация», на которой расположены поля для ввода логина (Email) и пароля, а так же кнопки «войти» - рисунки 23-24.

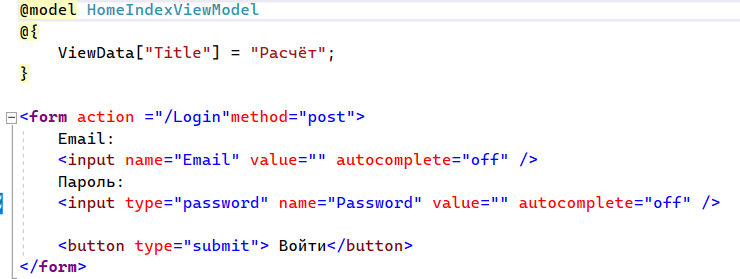


Рис. 23 – поля для ввода Email и пароля

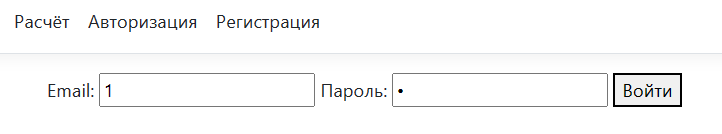


Рис. 24 – интерфейс вкладки «Авторизация»

В папке Registr содержится файл Index, который представляет собой страницу «Регистрация», на которой расположены поля для ввода логина (Email) и пароля, а так же кнопки «зарегистрироваться» - рисунки 25-26.

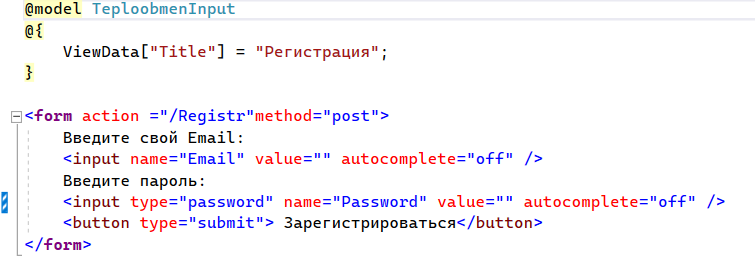


Рис. 25 – поля для регистрации пользователя

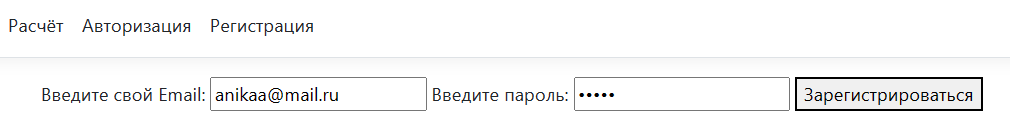


Рис. 26 – интерфейс вкладки «Регистрация»

В папке User содержится файл Index, который представляет собой страницу «личный кабинет», на которой расположена надпись «Привет, \*почта юзера\*!» - рисунок 27.

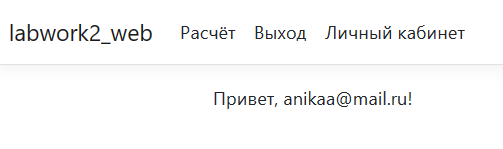


Рисунок 27 – интерфейс вкладки «Личный кабинет»

Ссылка на репозиторий: <https://github.com/anka1797/MPiSvM_2.git>