Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Информатика и информационные технологии»

Направление подготовки/ специальность: Автоматизированные системы обработки информации и управления

ОТЧЕТ

по проектной практике

Студент: Серебрякова Ирина Сергеевна, группа 241-339

Место прохождения практики: Московский Политех, кафедра «Информатика и информационные технологии»

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: Меньшикова Н.П

Москва 2025

**Оглавление**

[Введение 3](#_Toc198797618)

[Общая информация о проекте 4](#_Toc198797619)

[Название проекта 4](#_Toc198797620)

[Цель проекта 4](#_Toc198797621)

[Общая характеристика деятельности организации 4](#_Toc198797622)

[Описание задания по проектной практике 5](#_Toc198797623)

[Базовая часть задания 5](#_Toc198797624)

[Вариативная часть задания 10](#_Toc198797625)

[Описание достигнутых результатов по проектной практике 14](#_Toc198797626)

[Заключение 15](#_Toc198797627)

Введение

Предметом настоящего отчёта является описание хода и результатов проектной практики, предусмотренной учебным планом и направленной на практическую реализацию теоретических знаний и компетенций, полученных в процессе обучения. Целью практики являлась разработка функционального веб-сайта для Объединённой первичной профсоюзной организации Московского политехнического университета.

В рамках реализации проекта мною было принято участие в создании рабочей версии веб-сайта, ориентированного на удовлетворение информационных и сервисных потребностей сотрудников и студентов университета. Выполненный комплекс работ включал в себя все этапы жизненного цикла веб-разработки: от проведения анализа целевой аудитории и проектирования архитектуры сайта до непосредственной вёрстки страниц, проведения тестирования и подготовки необходимой технической документации.

В ходе выполнения проекта применялись современные инструменты и технологии веб-разработки. Особое внимание уделялось сбору и систематизации нормативно-справочной информации, необходимой для наполнения сайта, разработке привлекательного и удобного пользовательского интерфейса, а также созданию интерактивных прототипов страниц с последующей их программной реализацией.

Прохождение данной проектной практики позволило получить мне ценный практический опыт в области веб-разработки, взаимодействия с командой и анализа требований пользователей. В настоящем отчёте представлено детальное описание процесса выполнения поставленных задач, полученных компетенций и достигнутых результатов в рамках проекта "Сервисы для профсоюзной организации".

Общая информация о проекте

Название проекта: Сервисы для профсоюзной организации

Цель проекта: Создание рабочей версии веб-сайта Профорганизации Московского Политеха, отвечающей запросам сотрудников и студентов университета

Задачи проекта:

1. Проведение опроса и анализ потребностей целевой аудитории (студентов и сотрудников).

2. Сбор и систематизация нормативно-справочной информации.

3. Разработка архитектуры сайта (карты сайта).

4. Разработка современного и привлекательного дизайна сайта.

5. Сбор и подготовка справочной информации (новости, события, отчеты).

6. Создание интерактивных макетов каждой страницы сайта.

7. Верстка (программирование) страниц сайта.

8. Тестирование функциональности и удобства использования сайта.

9. Сбор и анализ обратной связи от заказчика и целевой аудитории для внесения корректировок.

10. Разработка технической документации.

Общая характеристика деятельности организации

Заказчиком проекта выступает Объединенная первичная профсоюзная

организация Московского политехнического университета Московской

городской организации общероссийского профсоюза образования.

Профсоюз защищает права студентов и обучающихся, является оператором

выплат, таких как - стипендии, материальные помощи, дотации. Профсоюзная

организация — это возможности, которые может получить любой человек,

состоящий в профсоюзе. Этими возможностями могут быть образовательные

форумы, оздоровительные выезды, поездки на отдых, экскурсионные поездки.

На учёте в объединенной первичной профсоюзной организации состоят как

работники, так и обучающиеся.

Сайт нужен профсоюзной организации для: поддержания имиджа бренда, повышения вовлеченности сотрудников и студентов о профсоюзной деятельности, удобства и доступности информации, привлечения новых членов профсоюза, документирования и отчетности.

Описание задания по проектной практике

Базовая часть задания

Цель: создание статического сайта на HTML и CSS, посвящённого проектной деятельности команды, с разделами: главная страница, информация о проекте, команда участников, журнал событий и страница ресурсов.

Этапы выполнения:

1. Планирование структуры сайта

• Определено, что сайт будет включать 5 основных страниц: index.html, about.html, team.html, journal.html, resources.html.

• Принята общая визуальная концепция: светлый текст на тёмном полупрозрачном фоне, использование единого фона для всех страниц, акцент на удобочитаемость.

2. Верстка и оформление

• Создан базовый шаблон на HTML5.

• Оформлен header с навигацией по сайту.

• Использован CSS для стилизации: фоновое изображение, оформление кнопок, блоков, списков и изображений.

3. Реализация страницы ресурсов (resources.html)

• Добавлены ссылки на полезные статьи и материалы.

• Вставлены изображения диаграммы Ганта и дорожной карты проекта.

• Оформлены подписи и стилизованные блоки с фоном и радиусами границ.

4. Адаптация под разные экраны

• Использовано max-width для изображений и текстовых блоков.

• Весь сайт корректно отображается на различных устройствах (ПК, планшет, мобильный).

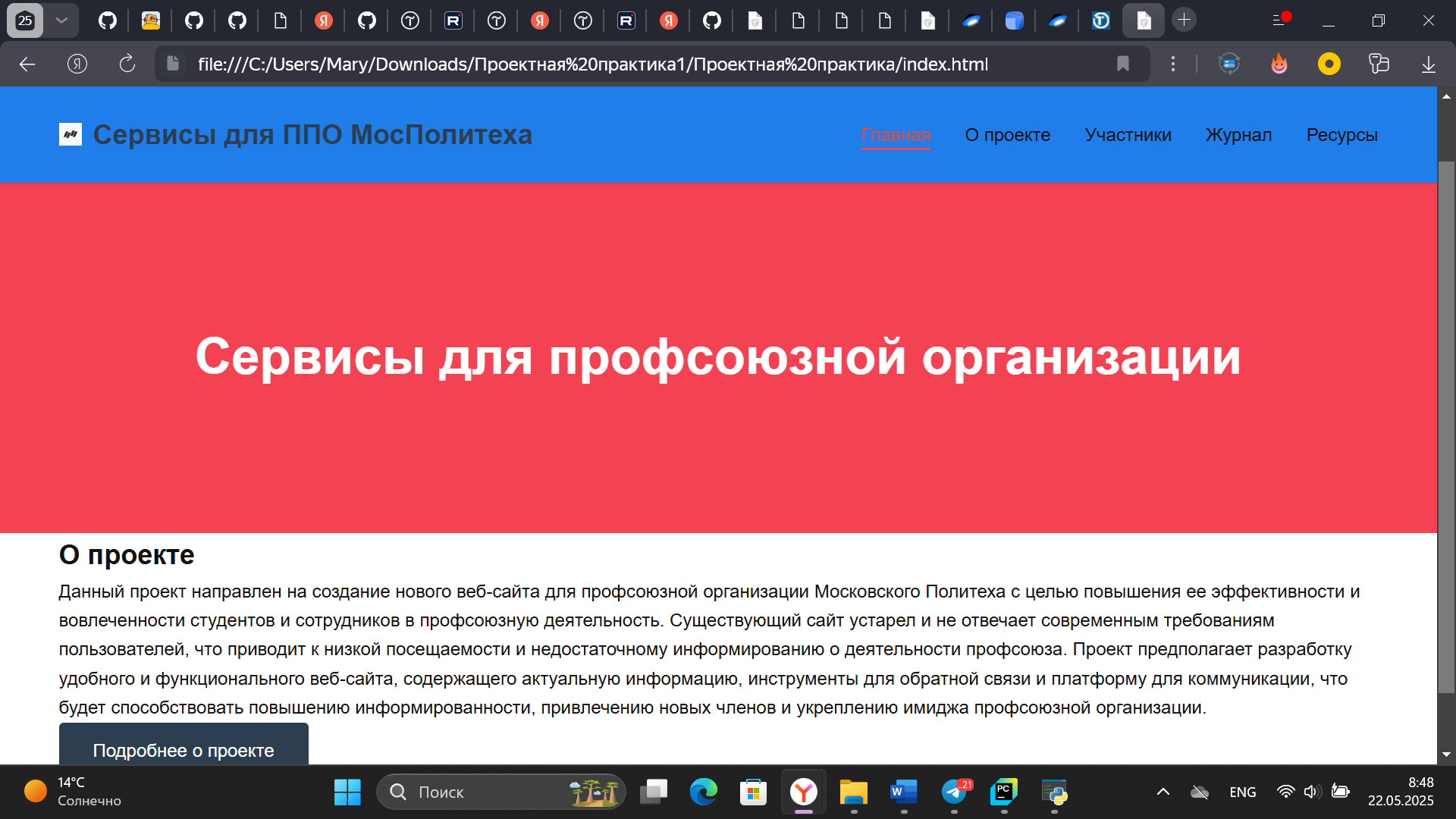
5. Завершение и тестирование

• Проверена работоспособность навигации между страницами.

• Убедились, что все ссылки и изображения загружаются корректно.

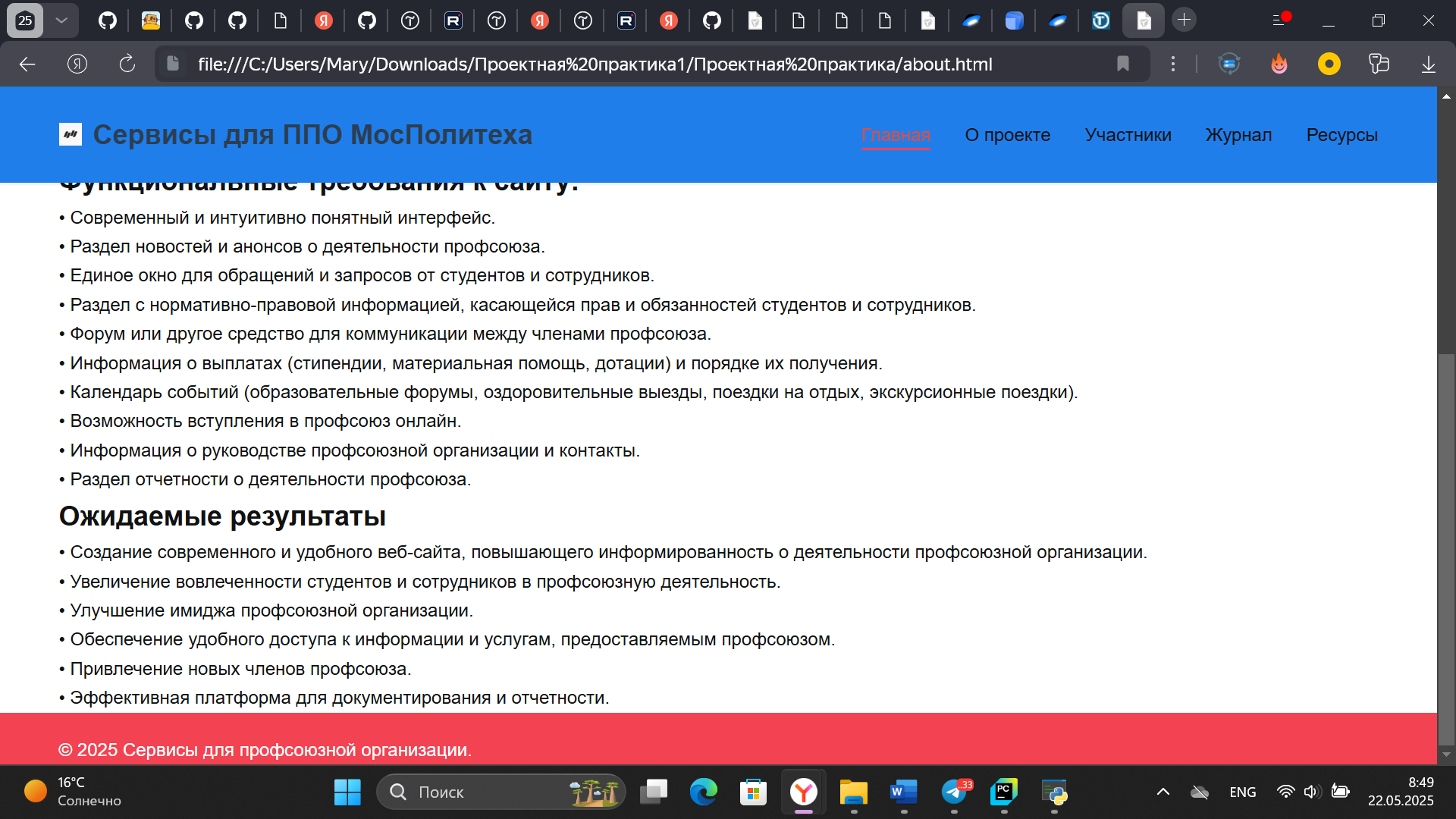
• Проведён финальный визуальный осмотр и правка недочётов.

Главная страница:

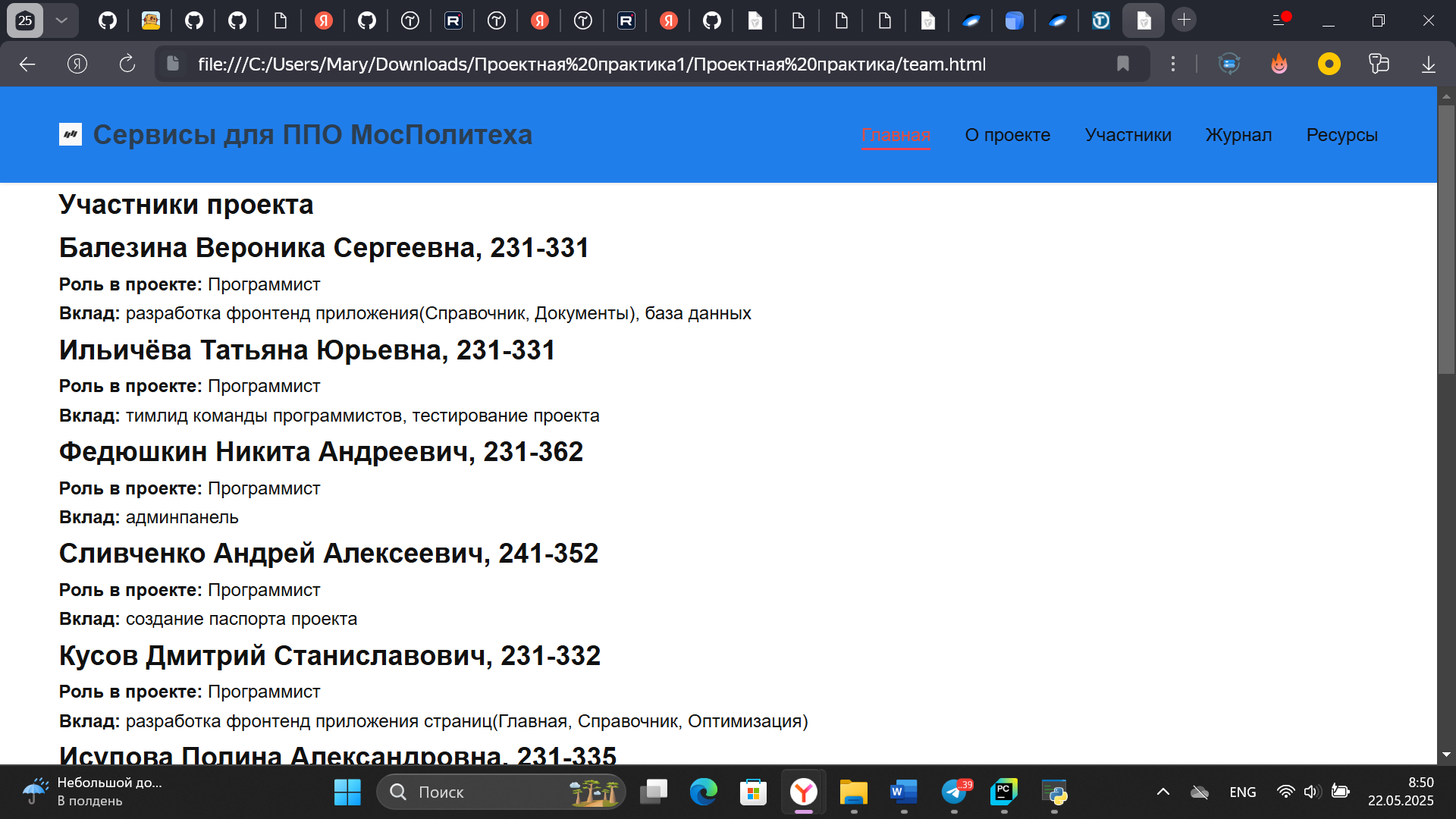


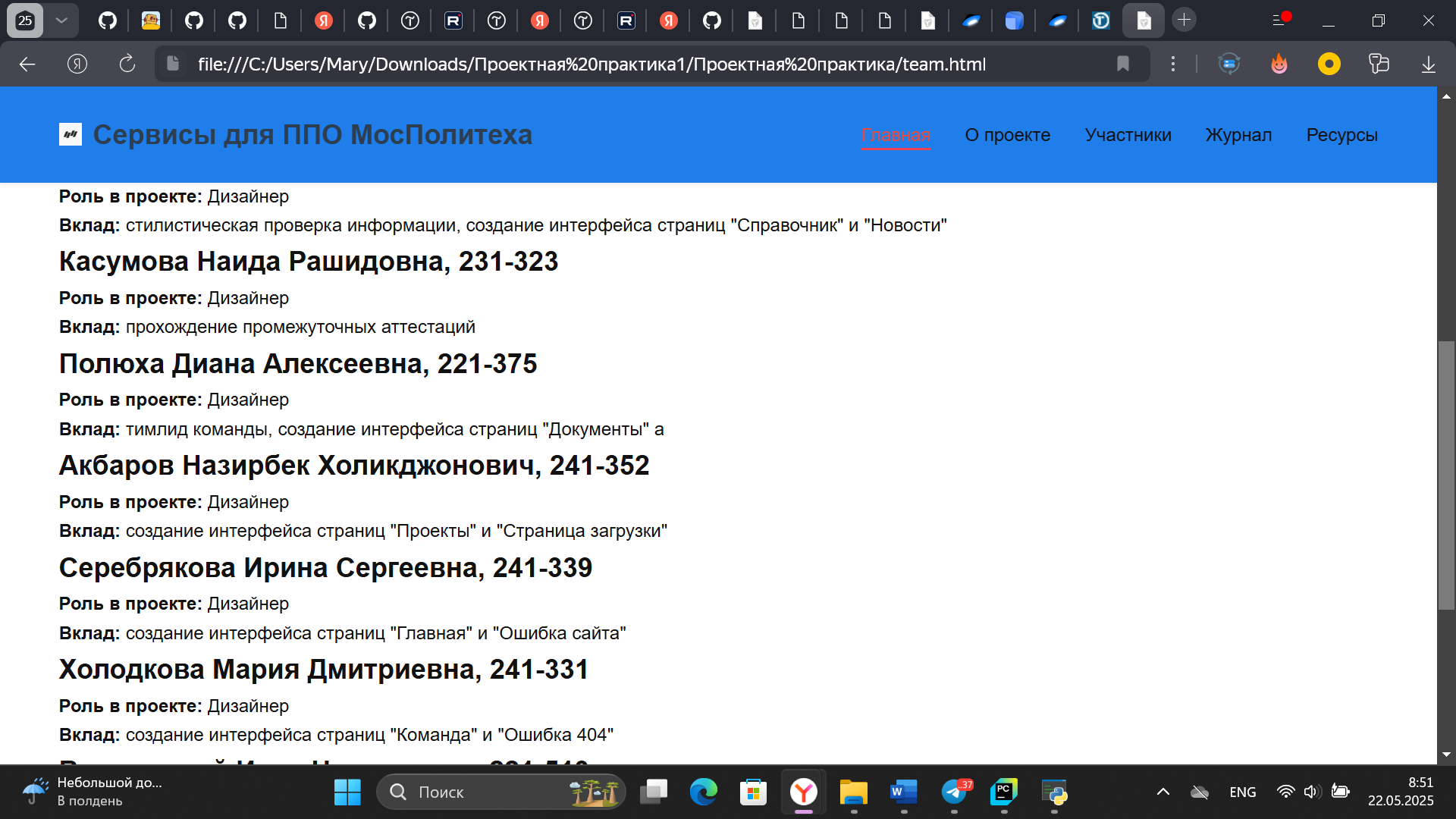
О проекте:

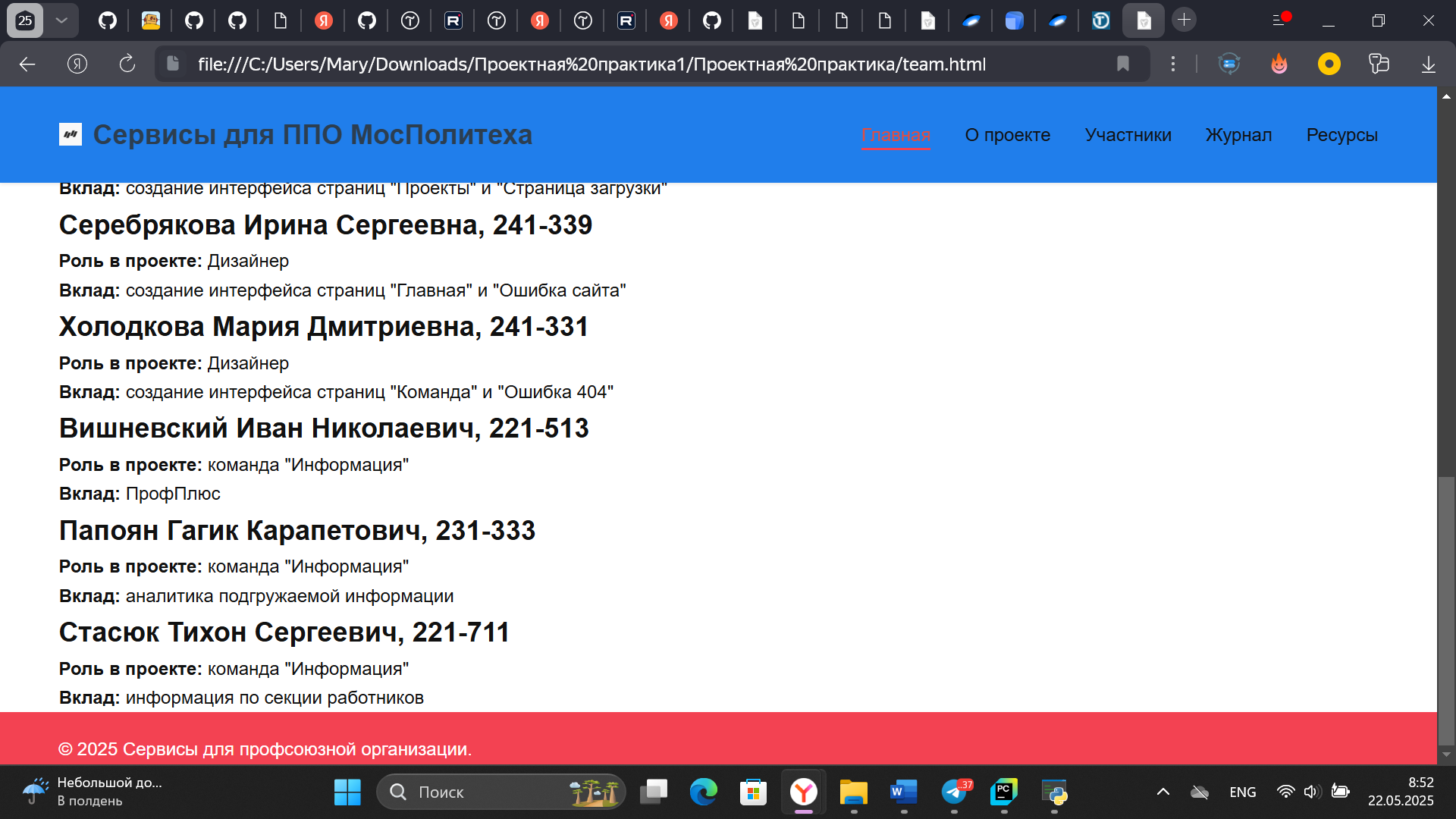




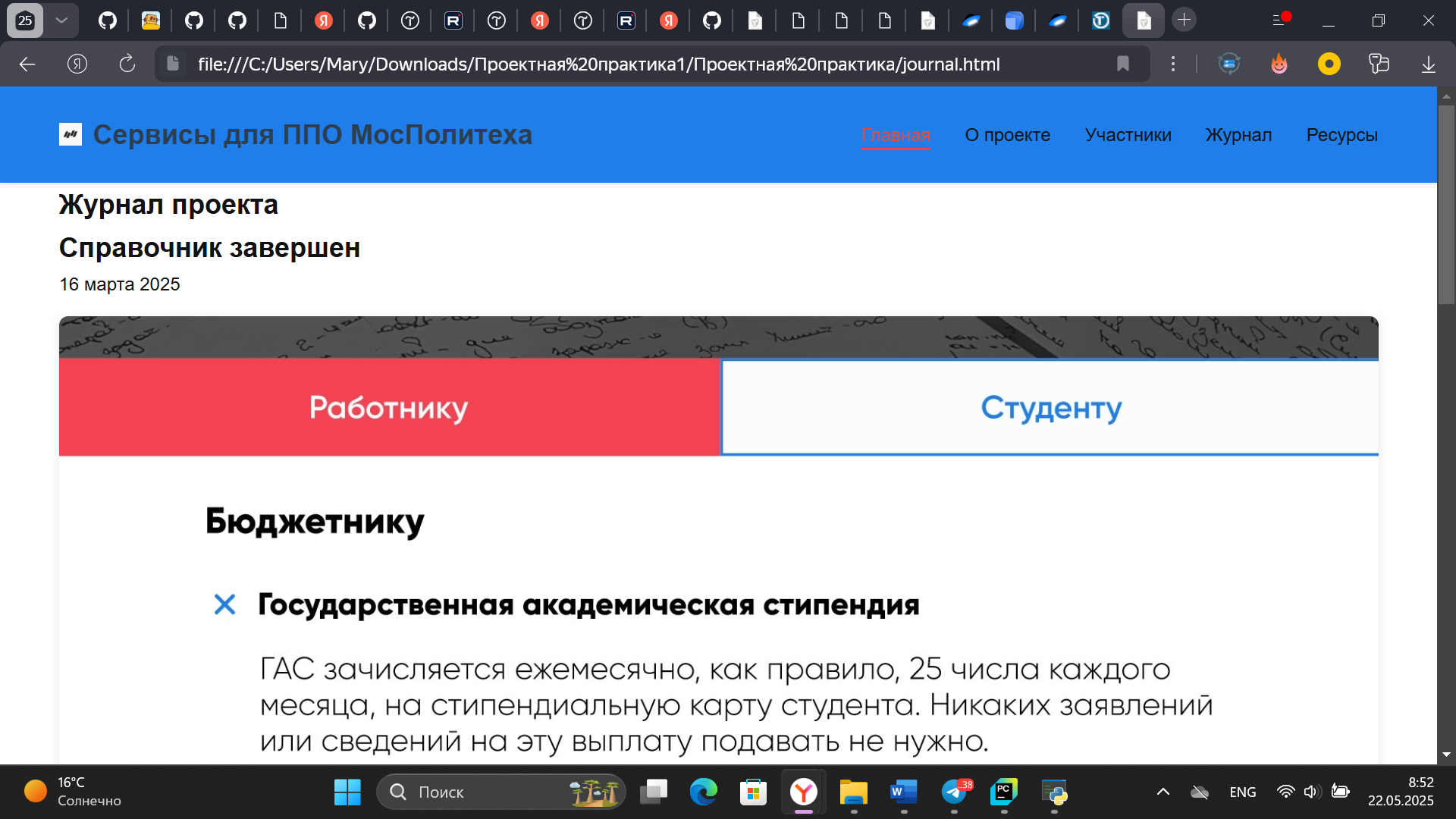
Участники:

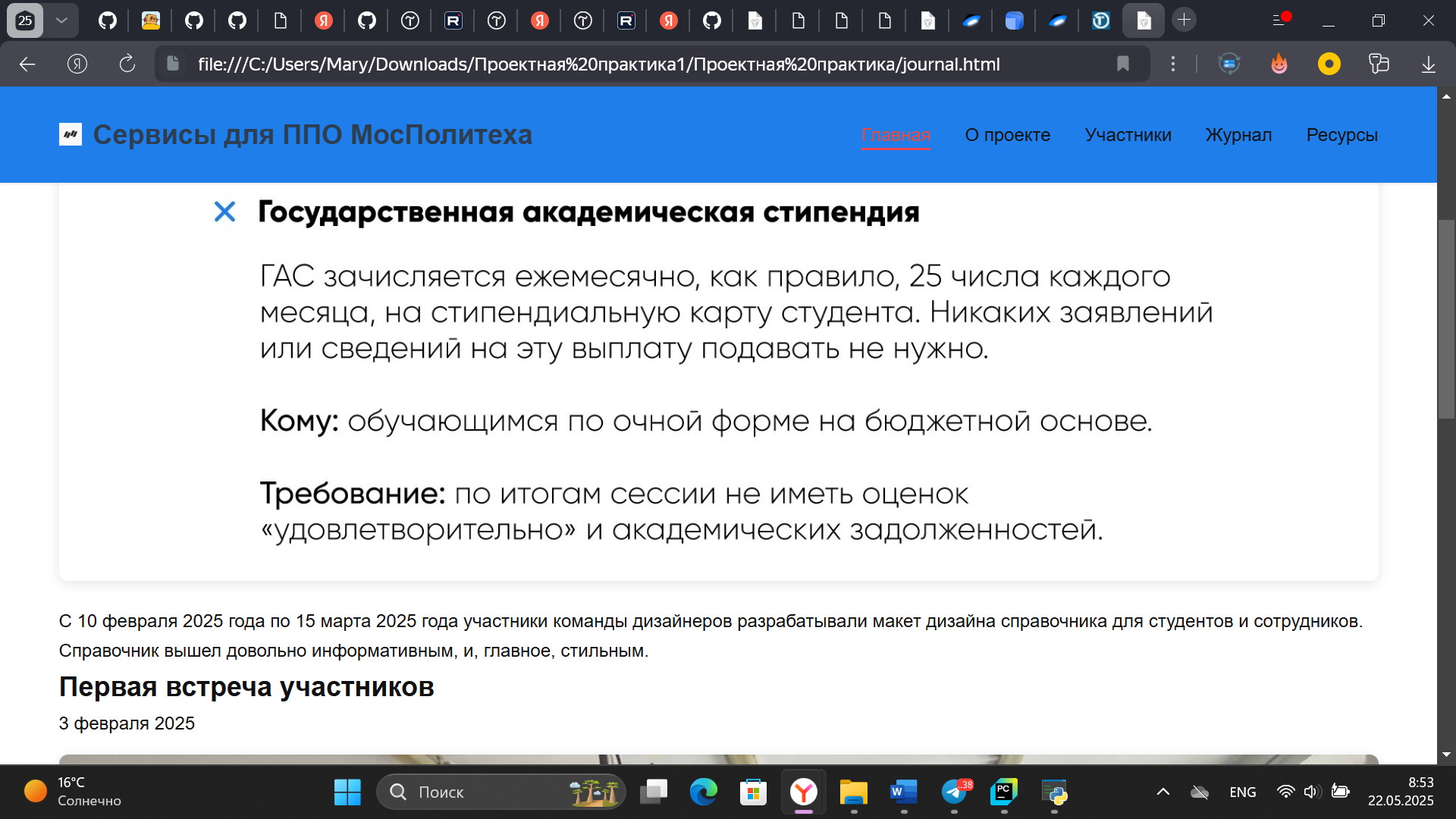


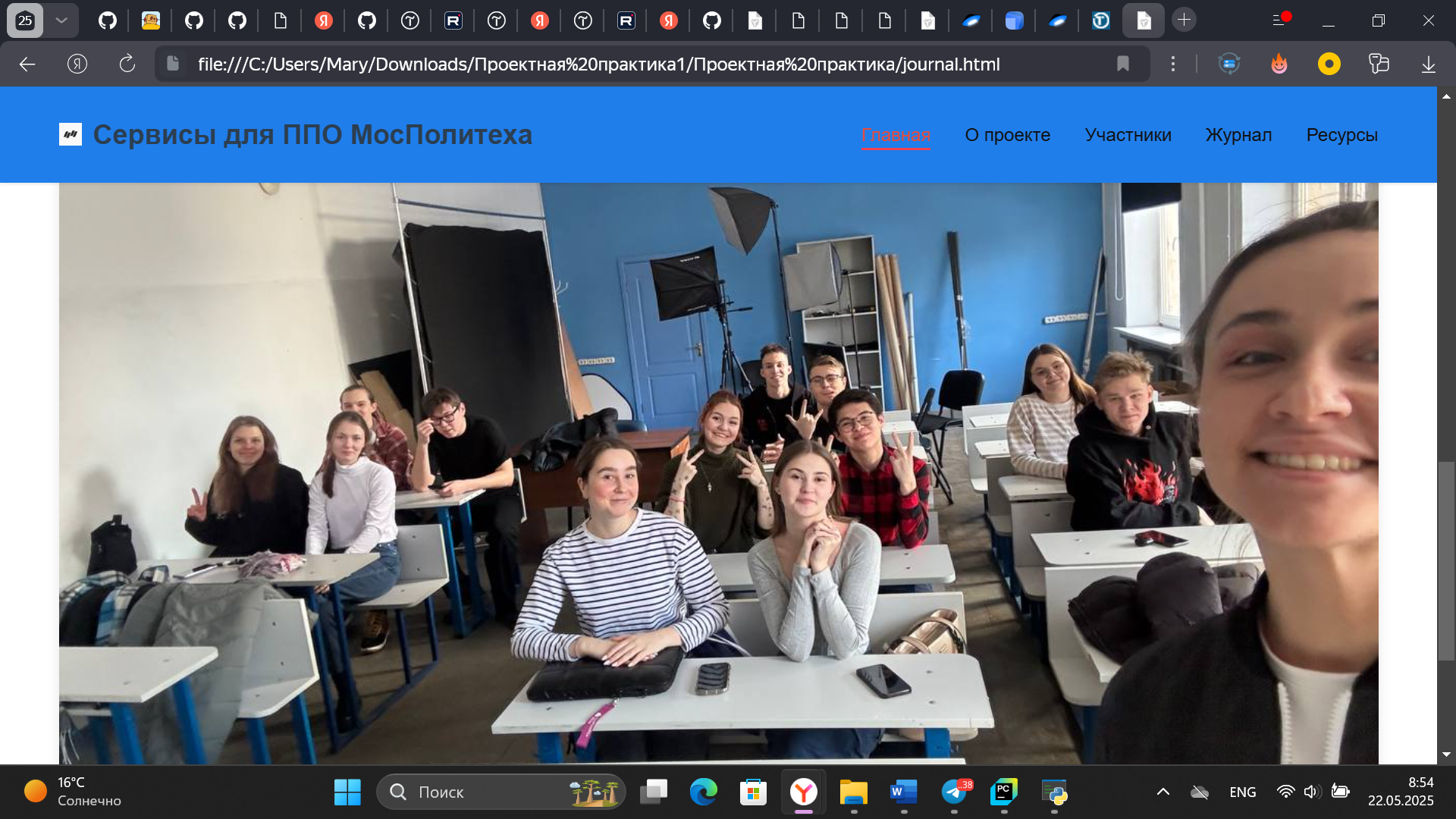




Журнал:

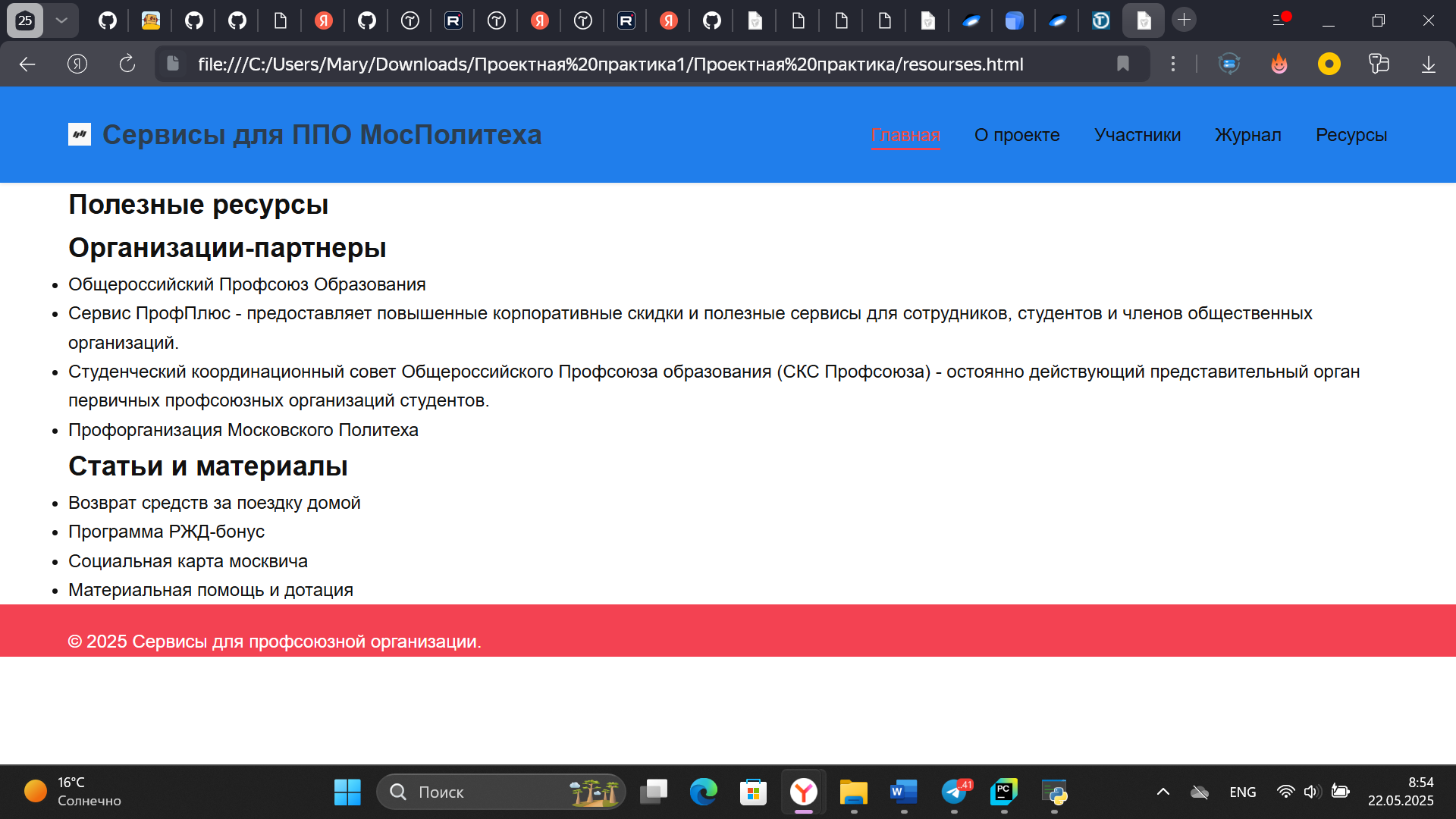








Ресурсы:



Итог: в рамках проекта был разработан и реализован готовый к использованию статический многостраничный веб-сайт, который всесторонне отражает цели, задачи и достигнутые результаты нашей команды. Структура сайта включает в себя информационные разделы, визуальные компоненты и интерактивные элементы в виде активных ссылок

Вариативная часть задания

Тема: Создание Telegram-bot’а на языке программирования Python

Цель проекта: создать Telegram-bot’ на Python, позволяющего

* Узнать информацию об основных персонажах кино- и комикс- вселенной Marvel.
* Изучить информацию о самых культовых фильмах и сериалах киновселенной Marvel.
* Ознакомиться с порядком просмотра фильмов, так как вселенная достаточно обширная и разобраться самостоятельно порой бывает сложно.

Порядок выполнения работы:

1. Первым шагом необходимо было изучить библиотеки python, с помощью которых можно создавать бота. Для работы была выбрана наиболее подходящая библиотека: python-telegram-bot. Далее её необходимо было установить. Через командную строку вводим pip install python-telegram-bot
2. Далее для работы нам нужно получить собственный токен для Telegram-bot. В боте @BotFather создаём бота и получаем собственный токен, далее он понадобится нам в коде.
3. Далее начинаем работать над ботом непосредственно в коде:
   1. Подключаем библиотеки:

import logging  
from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup  
from telegram.ext import Application, CommandHandler, CallbackQueryHandler, ContextTypes

* 1. Сохраняем наш токен в переменной с названием TOKEN
  2. Настраиваем логгирование:

logging.basicConfig(  
 format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',  
 level=logging.INFO  
)  
logger = logging.getLogger(\_\_name\_\_)

* 1. Вводим данные о фильмах, персонажах, комиксах:

data = {  
 "movies": {  
 "avengers\_endgame": {  
 "title": "Мстители: Финал",  
 "year": 2019,  
 "description": "Оставшиеся герои пытаются исправить последствия щелчка Таноса. Фильм содержит путешествия во времени, возвращение погибших персонажей и эпическую битву против Таноса. Завершает 22-фильмовую Сагу Бесконечности, принося жертвы (Железный человек, Чёрная вдова) и устанавливая новый статус-кво. Содержит множество отсылок к предыдущим фильмам франшизы.",  
 "image\_url": "https://avatars.mds.yandex.net/i?id=e729dcf1495aa58ad91fc470f8937945\_l-4338066-images-thumbs&n=13",  
 "required\_viewing": "Все фильмы Саги Бесконечности, особенно 'Мстители: Война бесконечности'"  
 } ,

#ввод остальных фильмов

"blackpanter": {  
 "title": "Чёрная пантера",  
 "year": 2018,  
 "description": "После смерти отца Т'Чалла возвращается в скрытую технологически развитую африканскую страну Ваканда, чтобы занять трон. Его правление оспаривает Эрик Киллмонгер, раскрывающий тёмные секреты прошлого. Фильм отличается уникальной культурной эстетикой, поднимает важные социальные вопросы и представляет вибраниум - самый прочный материал во вселенной.",  
 "image\_url": "http://images-s.kinorium.com/movie/poster/566968/w1500\_45947143.jpg",  
 "required\_viewing": "'Капитан Америка: Гражданская война' (первое появление Т'Чаллы)."  
 } ,  
 },  
 "characters": {  
 "iron\_man": {  
 "name": "Железный человек (Тони Старк)",  
 "description": "Гениальный изобретатель-миллиардер, создавший высокотехнологичный костюм. Основатель Мстителей.",  
 "films": [  
 "Железный человек (2008)",  
 "Железный человек 2 (2010)",  
 "Мстители (2012)",  
 "Железный человек 3 (2013)",  
 "Мстители: Эра Альтрона (2015)",  
 "Первый мститель: Противостояние (2016)",  
 "Человек-паук: Возвращение домой (2017, камео)",  
 "Мстители: Война бесконечности (2018)",  
 "Мстители: Финал (2019)"  
 ],  
 "image\_url": "https://www.nashe.ru/storage/62731/conversions/zGdBGpsXCq-social.jpg"  
 },

#описание остальных персонажей

"loki": {  
 "name": "Локи",  
 "description": "Бог обмана, брат Тора, антигерой с комплексом неполноценности.",  
 "films": [  
 "Тор (2011)",  
 "Мстители (2012)",  
 "Тор 2: Царство тьмы (2013)",  
 "Тор: Рагнарёк (2017)",  
 "Мстители: Война бесконечности (2018)",  
 "Локи (сериал, 2021)",  
 "Доктор Стрэндж: В мультивселенной безумия (2022, альтернативная версия)"  
 ],  
 "image\_url": "https://img.championat.com/s/1350x900/news/big/s/b/startovali-syomki-vtorogo-sezona-seriala-loki-s-tomom-hiddlstonom\_16545839271434379306.jpg"  
 }  
 },  
 "comics": {  
 "infinity\_gauntlet": {  
 "title": "Перчатка Бесконечности (1991)",  
 "description": "Культовая арка, где Танос собирает все Камни Бесконечности, становясь богом. Мстители и космические существа объединяются, чтобы остановить его.",  
 "characters": ["Танос", "Мстители", "Серебряный Сёрфер"],  
 "events": "Сага Бесконечности",  
 "image\_url": "https://static.wikia.nocookie.net/marvel/images/e/e1/Infinity\_Gauntlet\_Vol\_1\_1.jpg/revision/latest/scale-to-width-down/1200?cb=20200910120209&path-prefix=ru"  
 },  
 #вводим данные об стальных комиксах

"secret\_invasion": {  
 "title": "Тайное вторжение (2008)",  
 "description": "Скруллы годами заменяли ключевых героев и политиков, готовя масштабное вторжение на Землю. Никому нельзя доверять.",  
 "characters": ["Капитан Марвел", "Железный человек", "Человек-паук"],  
 "events": "Secret Invasion",  
 "image\_url": "https://cdn.marvel.com/u/prod/marvel/i/mg/2/f0/511c078aa9163/standard\_incredible.jpg"  
 }  
}  
}

* 1. Прописываем функцию /start. Это начало работы нашего бота, то, что пользователь видит, когда впервые заходит в него:

async def start(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 *"""Главное меню"""* if update.message:  
 message = update.message  
 else:  
 message = update.callback\_query.message  
  
 keyboard = [  
 [InlineKeyboardButton("🎬 Фильмы", callback\_data='show\_movies')],  
 [InlineKeyboardButton("🦸 Персонажи", callback\_data='show\_characters')],  
 [InlineKeyboardButton("📚 Комиксы", callback\_data='show\_comics')],  
 ]  
 await message.reply\_text(  
 'Добро пожаловать в энциклопедию Marvel! Выберите категорию:',  
 reply\_markup=InlineKeyboardMarkup(keyboard)  
 )

* 1. Обрабатываем кнопки с выбором категорий:

async def handle\_button(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 *"""Обработчик кнопок"""* query = update.callback\_query  
 await query.answer()  
  
 try:  
 if query.data == 'back':  
 await start(update, context)  
 elif query.data == 'show\_movies':  
 await query.message.reply\_text("Вы выбрали категорию: 🎬 Фильмы")  
 await show\_category(query, "movies", "Фильмы")  
 elif query.data == 'show\_characters':  
 await query.message.reply\_text("Вы выбрали категорию: 🦸 Персонажи")  
 await show\_category(query, "characters", "Персонажи")  
 elif query.data == 'show\_comics':  
 await query.message.reply\_text("Вы выбрали категорию: 📚 Комиксы")  
 await show\_category(query, "comics", "Комиксы")  
 elif query.data.startswith('movie\_'):  
 await show\_item(query, "movies", query.data[6:])  
 elif query.data.startswith('character\_'):  
 await show\_item(query, "characters", query.data[10:])  
 elif query.data.startswith('comic\_'):  
 await show\_item(query, "comics", query.data[6:])  
 except Exception as e:  
 logger.error(f"Ошибка: {e}")  
 await query.message.reply\_text("Произошла ошибка. Попробуйте еще раз.")

Также было важно включить в этот раздел обработку ошибок для комфорта пользователя.

* 1. Обработка выбора категории:

async def show\_category(query, category, title):  
 *"""Показать меню категории"""* items = data[category]  
 keyboard = []  
  
 for item\_id in items:  
 item = items[item\_id]  
 btn\_text = item.get('name', item.get('title'))  
 callback\_data = f"{category[:-1]}\_{item\_id}"  
 keyboard.append([InlineKeyboardButton(btn\_text, callback\_data=callback\_data)])  
  
 keyboard.append([InlineKeyboardButton("🔙 Назад", callback\_data='back')])  
  
 await query.message.reply\_text(  
 text=f"Выберите {title.lower()}:",  
 reply\_markup=InlineKeyboardMarkup(keyboard)  
 )

* 1. Обработка выбора элементов в категории:

async def show\_item(query, category, item\_id):  
 *"""Показать информацию о выбранном элементе с картинкой"""* item = data[category][item\_id]  
 text = f"<b>{item.get('name', item.get('title'))}</b>"  
  
 if 'year' in item:  
 text += f" ({item['year']})"  
  
 text += f"\n\n{item['description']}"  
  
 # Добавляем список фильмов для персонажей  
 if category == 'characters' and 'films' in item and item['films']:  
 text += "\n\n<b>Появляется в фильмах:</b>\n"  
 text += "\n".join(f"• {film}" for film in item['films'])  
  
 # Добавляем рекомендации для фильмов  
 if category == 'movies' and 'required\_viewing' in item:  
 text += f"\n\n<u>Что посмотреть перед просмотром:</u>\n{item['required\_viewing']}"  
  
 # Добавляем информацию о персонажах и событиях для комиксов  
 if category == 'comics':  
 if 'characters' in item and item['characters']:  
 text += f"\n\n<b>Ключевые персонажи:</b>\n{', '.join(item['characters'])}"  
 if 'events' in item:  
 text += f"\n\n<b>События:</b> {item['events']}"  
  
 # Отправляем картинку с описанием  
 await query.message.reply\_photo(  
 photo=item['image\_url'],  
 caption=text,  
 parse\_mode='HTML'  
 )  
  
 # Кнопка назад  
 keyboard = [[InlineKeyboardButton("🔙 Назад", callback\_data=f'show\_{category}')]]  
 await query.message.reply\_text(  
 "Выберите действие:",  
 reply\_markup=InlineKeyboardMarkup(keyboard)  
 )

* 1. Настройка запуска бота в самой среде программирования:

def main():  
 try:  
 app = Application.builder().token(TOKEN).build()  
 app.add\_handler(CommandHandler('start', start))  
 app.add\_handler(CallbackQueryHandler(handle\_button))  
  
 logger.info("Бот запускается...")  
 app.run\_polling()  
 except Exception as e:  
 logger.error(f"Ошибка запуска: {e}")

Итоговый и полный вариант кода содержится в репозитории GitHub: <https://github.com/serebriannka/Practice2025/tree/main>

**Описание достигнутых результатов по проектной практике:**

*Базовая часть задания:*

Создан функционирующий сайт, содержащий полную информацию о проекте «Сервисы для первичной профсоюзной организации МосПолитеха». Сайт отвечает требованиям, обозначенным в задании к базовой части, а так же соответствует стилистике самого сайта, который команда разрабатывает для профсоюзной организации МосПолитеха (учтена цветовая гамма: белый, синий и красный – цвета сайта и профсоюзной организации).

*Вариативная часть задания:*

Создан Telegram-bot на Python. Он соответствует заданию на эту часть и полностью функционирует, содержит в себе достаточно большой объём информации.