Министерство образования Республики Молдова

Технический университет Молдовы

Факультет Вычислительной техники, Информатики и Микроэлектроники

Программирование в сети

Отчет по лабораторной работе №3

Тема: «Protocol și aplicație client HTTP»

Выполнил Коваль Роман

ст. гр. TI-196:

Проверила: Ангилоглу Анна

Кишинев 2022

**Теория**

**HTTP —** это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (web-browser). Полученный итоговый документ будет (может) состоять из различных поддокументов, являющихся частью итогового документа: например, из отдельно полученного текста, описания структуры документа, изображений, видео-файлов, скриптов и многого другого.

Рисунок 1 – схема потоков в приложении



Клиенты и серверы взаимодействуют, обмениваясь одиночными сообщениями (а не потоком данных). Сообщения, отправленные клиентом, обычно веб-браузером, называются **запросами**, а сообщения, отправленные сервером, называются **ответами**.

Для разграничения действий с ресурсами на уровне HTTP-методов и были придуманы следующие варианты запросов:

* GET — получение ресурса
* POST — создание ресурса
* PUT — обновление ресурса
* DELETE — удаление ресурса

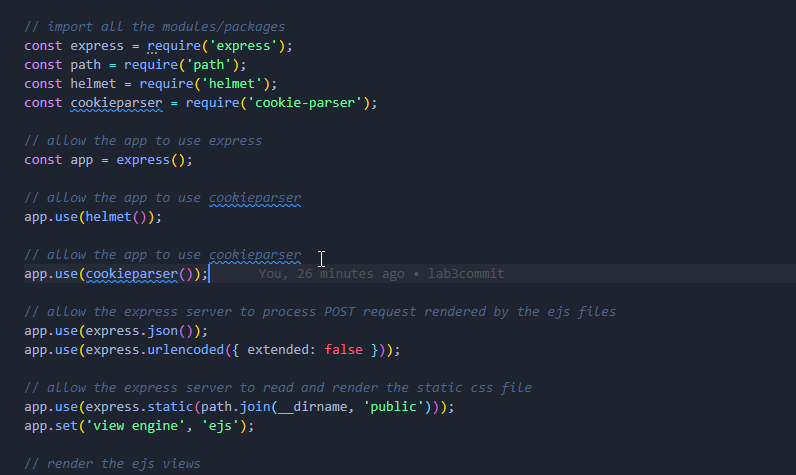
.

**Ход работы:**

**Постановка задачи**

Реализовать запросы HTTP клиент и POST-GET связи в нём. В качестве платформу использовать форму авторизации. Пользователь должен иметь возможность авторизоваться, после чего сервер сохраняет cookie ресурсы о текущем пользователе.

Рисунок 1 – фрагмент кода из index.js



Здесь происходит импорт необходимых для реализации клиента компонентов в файлы проекта и начинается их настройка.

Рисунок 2 – фрагмент кода из login.ejs

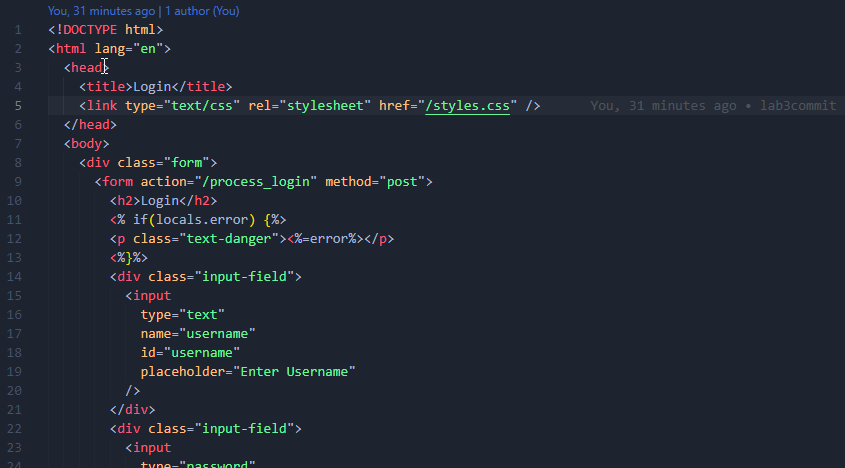
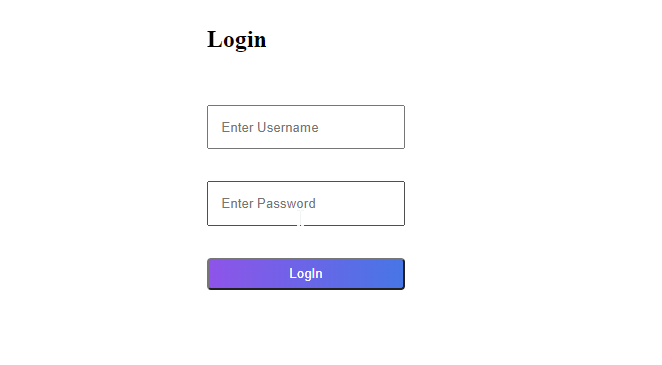
Выше можно видеть фрагмент кода из файла, который описывает страницу нашего логгинга, некоторую логику, а так же связывает её с визуалом из styles.css.

Рисунок 3 – фрагмент кода из styles.css

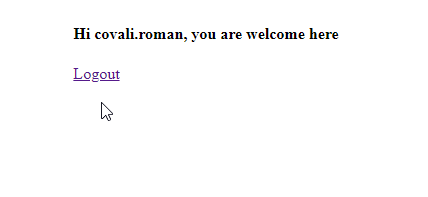


Рисунок 4 – визуал страницы логгинга



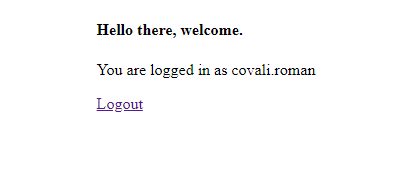
На этой странице пользователь вводит данные и инициирует первые POST и GET запросы. POST – запрос для добавление информации на сервер. GET – запрос для проверки данных и отображения страницы welcome.

Рисунок 5 – визуал сраницы welcome



После успешной авторизации пользователь получает следующую страницу, на которой указан его логин и присутствует кнопка выхода.

Рисунок 6 – главная страница



Теперь, если перейти на главную страницу после процедура авторизации, мы получим уже не страницу с логином, а страницу с сообщением о том, что пользователь залогинился под конкретным именем и информация об этом сохранена.

Рисунок 7 – запрос POST при авторизации, для отправки данных на сервер

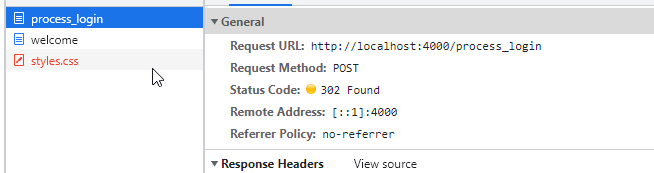
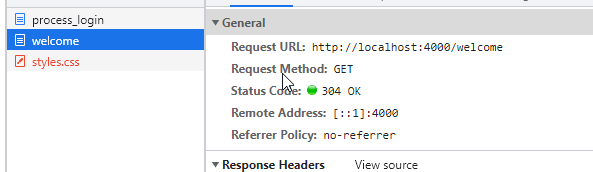


Рисунок 7 – запрос GET после авторизации, для отображение приветственной страницы



**Вывод:**

HTTP — это основа всего интернета. Он быстрый, легкий и многофункциональный. Подводя итоги, рассмотрим преимущества и особенности HTTP-протокола.

* **Скорость передачи**. Веб-страница содержит разные элементы, такие как текст и изображения. Для каждого элемента требуется разное количество ресурсов для хранения и загрузки. HTTP позволяет нескольким соединениям загружать отдельные элементы одновременно, тем самым ускоряя передачу данных.
* **Гибкость протокола**. Клиент всегда знает, какой тип файла нужно будет загрузить. Благодаря этому приложение на стороне клиента может быстро загрузить расширения или модули, если для отображения данных необходимы дополнительные ресурсы. Так, например, это могут быть Flash-плееры или программы чтения PDF-документов.
* **Безопасность соединения**. HTTP 1.0 загружает каждый файл через независимое соединение, а затем закрывает его. Это снижает риск перехвата данных во время передачи..
* **Легкость программирования**. HTTP закодирован в виде обычного текста, поэтому его легче реализовать, чем протоколы, которые используют код. Данные форматируются в виде текста, а не строк переменных или полей.
* **Возможности поиска**. Хотя HTTP — простой протокол обмена сообщениями, он позволяет искать информацию в базе данных с помощью одного запроса.Можно использовать протокол для выполнения SQL-поиска и возврата результатов, отформатированных в HTML-документе.