МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

ПРАКТИЧЕСКАЯ РАБОТА №3

по дисциплине «Проектирование и разработка серверных частей интернет-ресурсов»

Студент группы ИКБО-21-23 Муравьев А.О.

_____

(подпись студента)

Руководитель практической работы Благирев М.М.

_____

(подпись руководителя)

Работа представлена «___»_____2025 г.

Допущен к работе «___» _____2025 г.

Москва 2025

СОДЕРЖАНИЕ

ЦЕЛЬ РАБОТЫ

В задании предлагается создать сложную серверную конфигурацию, состоящую из связки apache+nginx+C#+База данных. В данной конфигурации предполагается создание как минимум 3 элементов(контейнеров) или использование как основы серверной конфигурации, созданной в практической работе №1. В этой конфигурации предполагается акселерированное проксирование без кэширования. Схематично предполагаемый алгоритм работы изображен на рисунке.

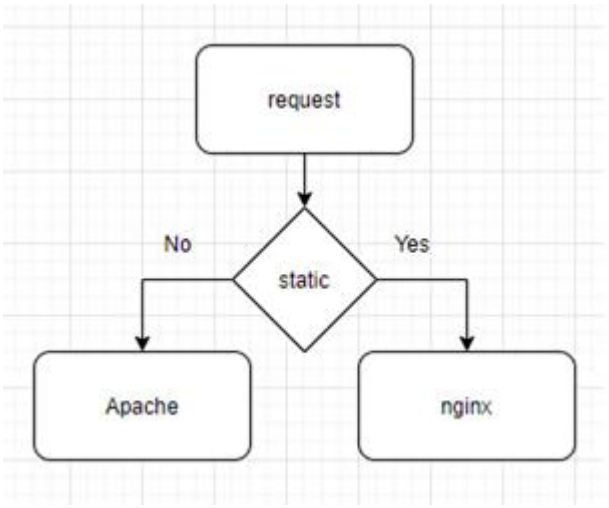

Рисунок 1 – Схематичное изображение предполагаемой функциональности

Предполагается, что сервер nginx будет отображать статический контент, а apache динамический и в связке мы получим быстродейственную и эффективную систему.

Также по необъяснимым обстоятельствам следует изменить root-директории с базовых на другие.

Для доступа к администрированию предусмотреть базовую авторизацию и аутентификацию с применением httaccess и хранением пользователей в БД (без использования htpasswd). Для тестирования данной конфигурации предполагается создать тестовое веб-приложение на тему по варианту: Кофейня.

Тестовое веб-приложение предполагает создание как минимум 2 вебстраниц со статическим контентом и двух веб-страниц с динамическим контентом: взятом из базы данных, например.

ХОД РАБОТЫ

Структура веб-приложения

Веб-приложение будет состоять из пяти страниц. Двух статических (главная страница и страница «О нас»). Двух динамических страниц (страница заказа с меню и страница всех заказов). И одной POST-страницы (страница после отправки заказа).

Реализована предложенная структура веб-приложения (рис. 1).

Nginx обрабатывает запросы, возвращает статические страницы и дёргает Apache при запросе динамики.

Apache устанавливает root-директорию, CGI-директорию, обрабатывает авторизацию на сервере (HTTP Basic Auth + PostgreSQL). Управляет доступом.

CGI-приложение, написанное на C#, обрабатывает GET и POST-запросы. Реализует динамику сайта.

База данных на PostgreSQL хранит в себе данные.

Итого получается 3 контейнера:

1. Nginx,
2. Apache + CGI C#,
3. PostgreSQL.

Реализация веб-приложения

Nginx обрабатывает пути. Если путь состоит из «/» или «/about», то он загружает статические файлы index.html или about.html соответственно (root директорию загружает сам Apache).

В случае столкновения с путями «/menu», «/order» (точное совпадение) или «/admin/» (частичное совпадение) Nginx дёргает Apache и создаёт GET-запрос для CGI-скрипта.

Схематичное представление работы конфигурации Nginx показано на рисунке 2.

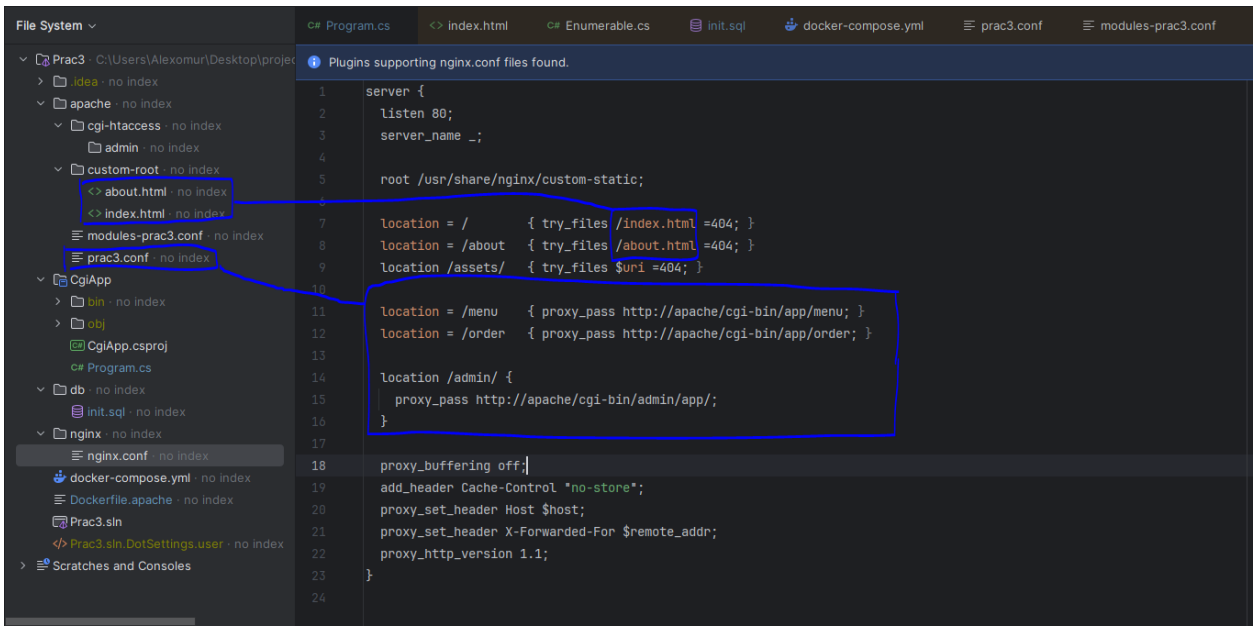

Рисунок 2 – Схематичное представление работы конфигурации Nginx

Конфигурация Apache отвечает за загрузку статической/корневой (root) директории, загрузку CGI, настройки авторизации (и связанной с авторизацией CGI) и загрузку PostgreSQL.

Конфигурация Apache (prac3.conf) представлена на рисунке 3.

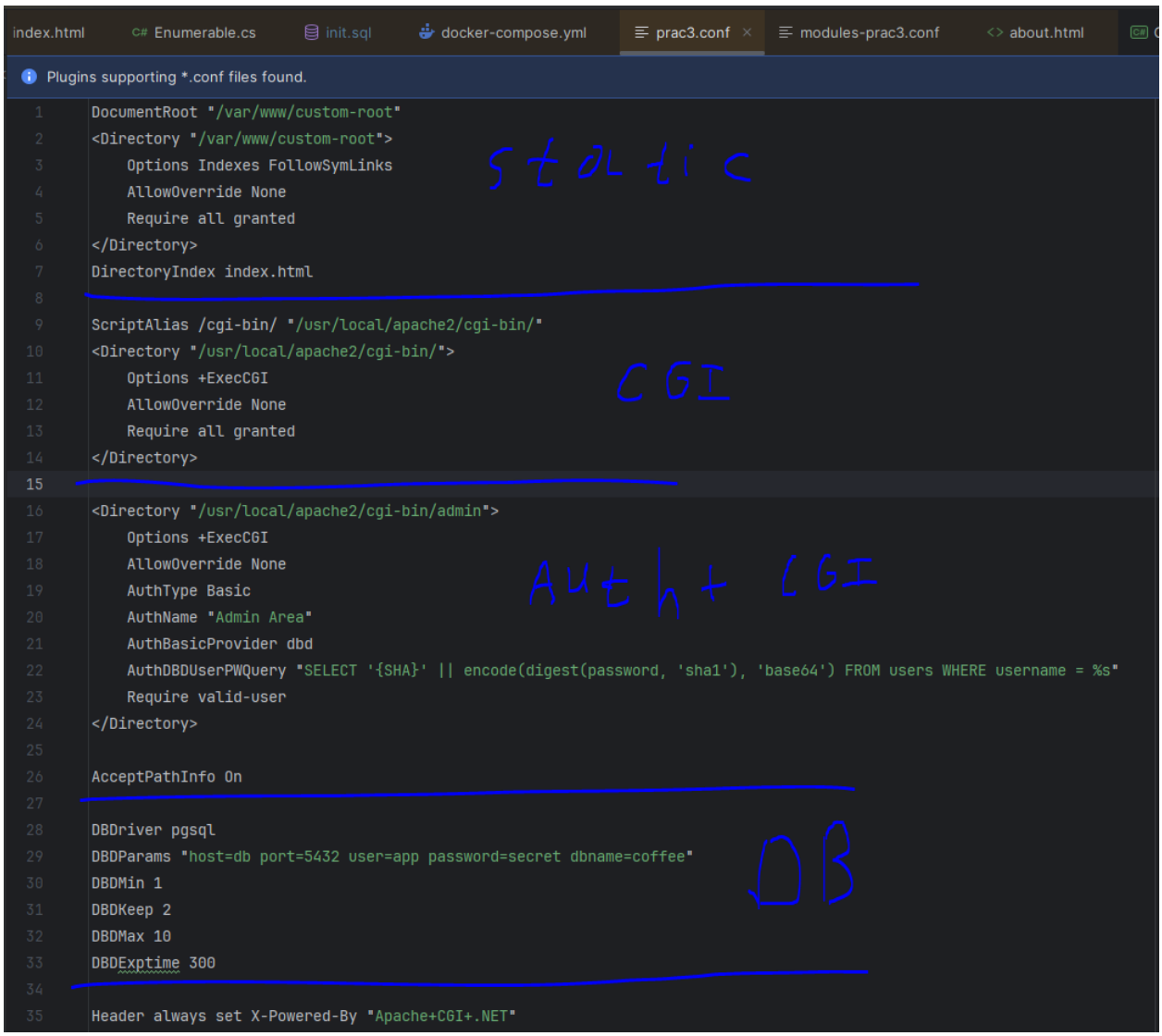

Рисунок 3 – Конфигурация Apache

CGI C# скрипт запускается Apache. Apache загружает все нужные данные (метод запроса, путь, другие данные о запросе) в переменные окружения.

В C# скрипте необходимо прописать всю логику любых запросов. В том числе секретную.

В данном случае авторизацией полностью занимается Apache + PostgreSQL. Запрос с админским доступом приходит только, если была пройдена авторизация в Apache.

Фрагмент кода (отвечающего за обработку POST-запроса) представлен на рисунке 4.

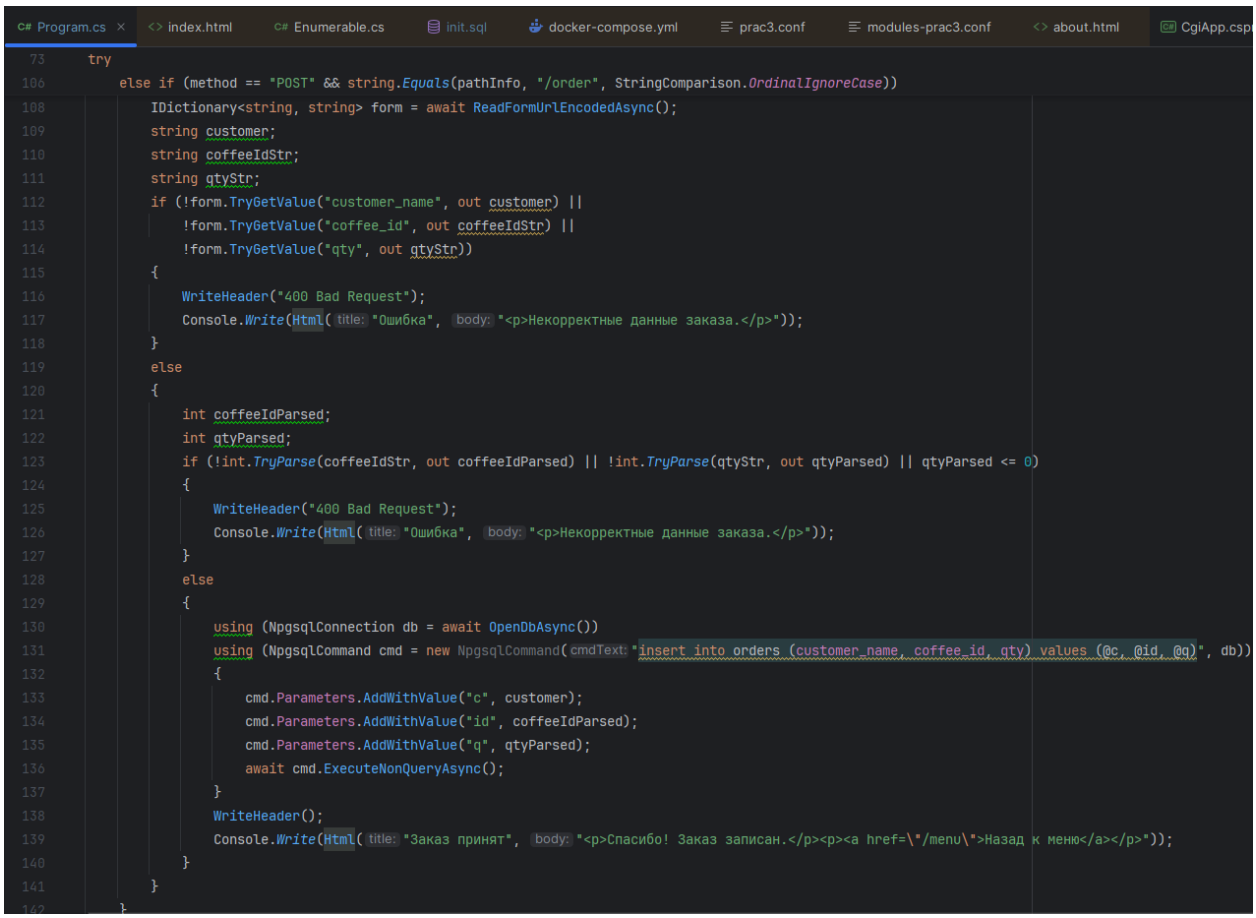

Рисунок 4 – Фрагмент CGI C# скрипта

Для корректной работы программы необходимо заполнить базу данных некой структурой и какими-то данными. Фрагмент init-sql-скрипта представлен на рисунке 5.

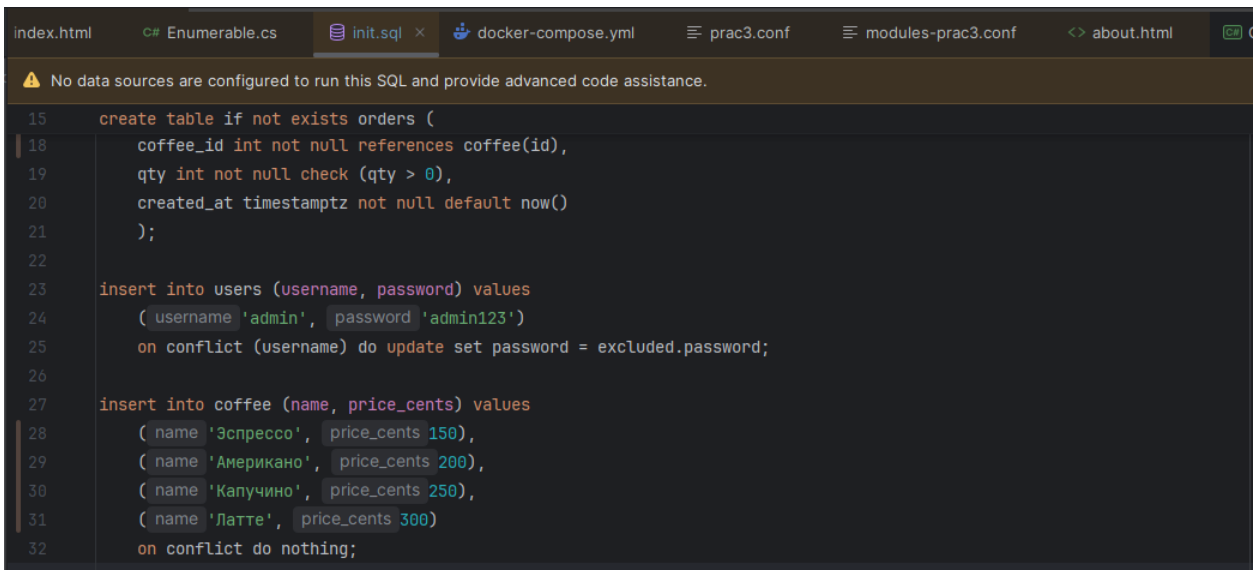

Рисунок 5 – Фрагмент init-sql-скрипта

Результат работы программы

Результат открытия пути «/» показан на рисунке 6.

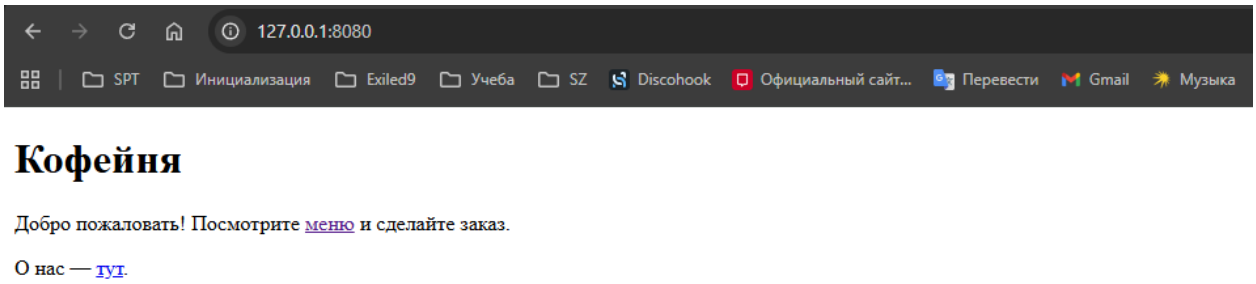

Рисунок 6 – Результат открытия пути «/»

На рисунке 7 показан результат открытия пути «/about».

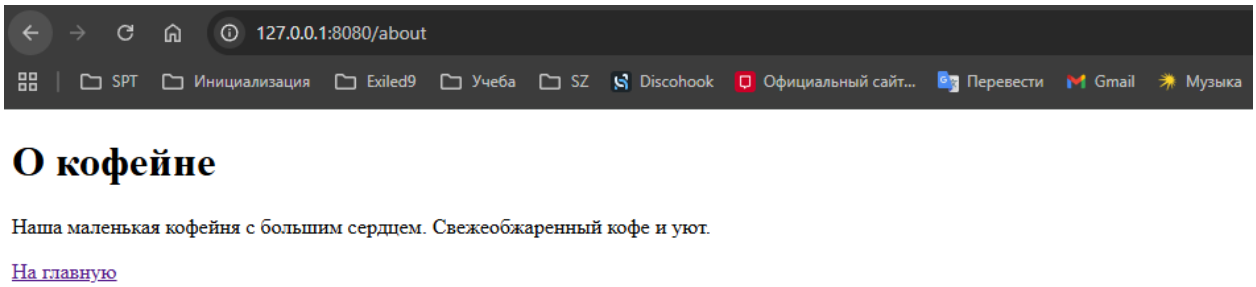

Рисунок 7 – Результат открытия пути «/about»

На рисунке 8 представлено открытие пути /menu (динамическая страница).

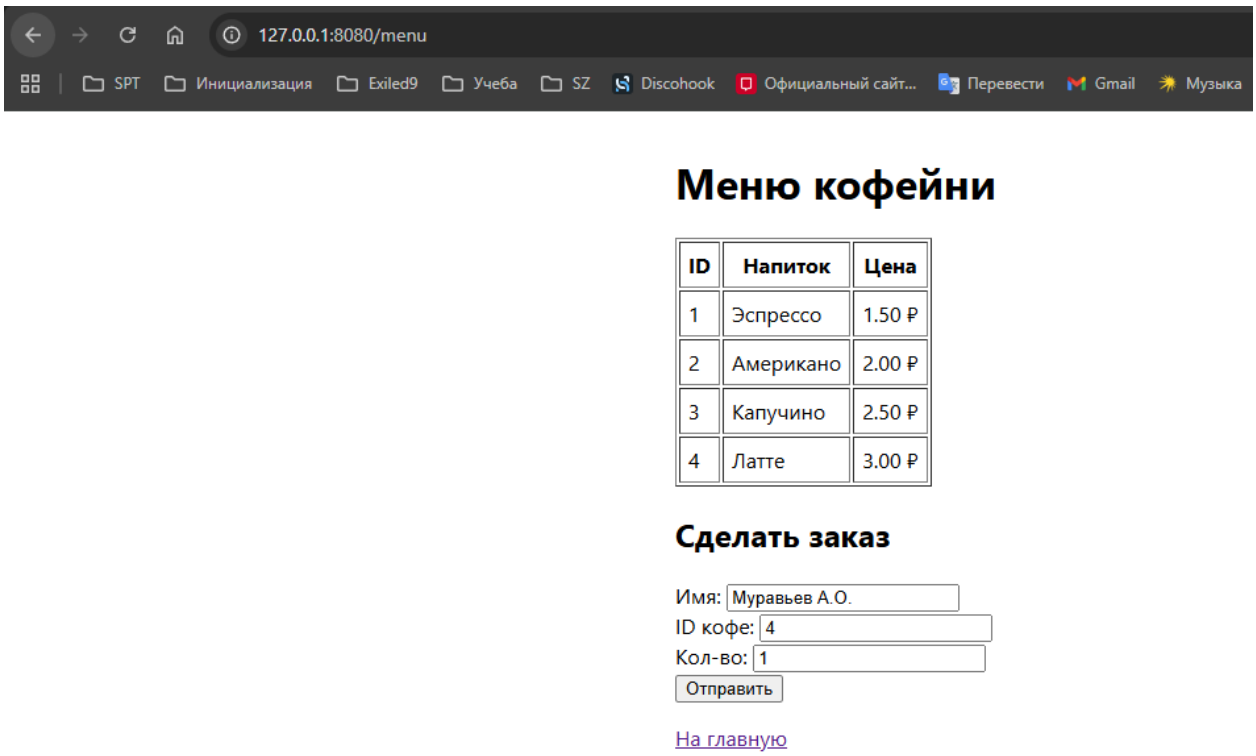

Рисунок 8 – Результат открытия пути «/menu»

На рисунке 9 представлен результат отправки POST-запроса с информацией о заказанном кофе.

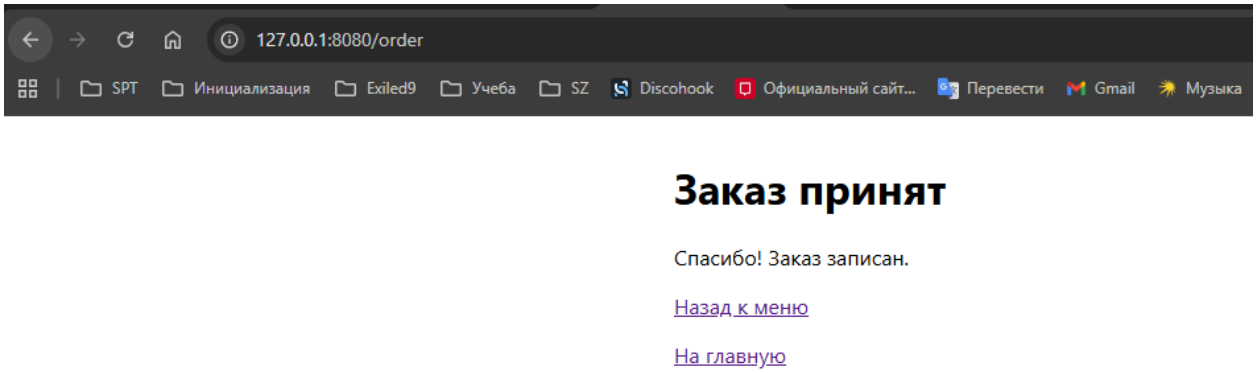

Рисунок 9 – Результат отправки POST-запроса

На рисунке 10 представлено окно авторизации при попытке войти в «Admin Area».

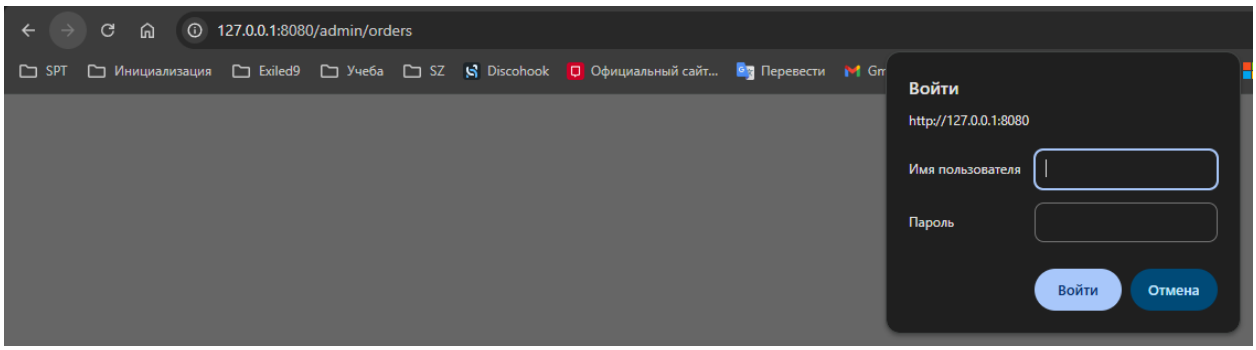

Рисунок 10 – Окно авторизации

На рисунке 11 представлена динамическая страница с последними заказами.

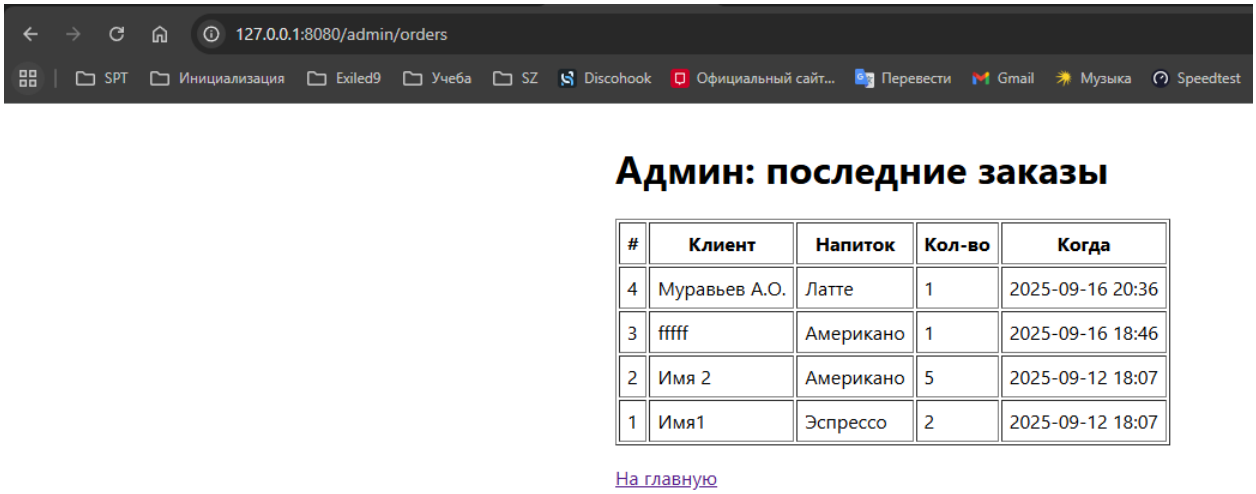

Рисунок 11 – Динамическая страница с данными о последних заказах

ВЫВОД

Таким образом, была разработана сложная серверная конфигурация, состоящая из связки Apache+CGI C#+Nginx+PostreSQL.

Исходный код проекта расположен по адресу:

https://github.com/alexomur/MireaBackend/tree/master/Prac3