**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет Инфокоммуникационных Технологий

Направление подготовки

«Программирование в Инфокоммуникационных Системах»

Лабораторная работа №4

**«Основы серверных приложений»**

Выполнил:

Фатин Алексей Андреевич

Группа №К3320

Проверила:

Марченко Елена Вадимовна

Санкт-Петербург

2024

# **Введение**

Цель**:** познакомиться с основами серверных приложений на PHP и Node.js

Задачи**:**

1. Разработать веб-приложение, позволяющее пользователю создавать заказы
2. Изменить скрипт WordPress так, чтобы при авторизации пароль пользователя сохранялся в таблицу БД в изначальном и инвертированных видах.
3. Создать веб-сервер, возвращающий статический index.html

# **Ход работы**

Задание 1. Веб-приложение на PHP

Было разработано приложение, запрашивающее у пользователя его фамилию, имя, отчество, адрес, телефон, электронную почту, а также предлагающее добавить комментарий к заказу и выбрать один из двух товаров (рис. 1).

A screenshot of a computer

Description automatically generated

Рисунок 1 – форма

При заполнении полей и нажатии на кнопку «Купить» данные сохраняются в базе данных (рис. 2).

A screenshot of a phone number

Description automatically generated

Рисунок 2 – данные в БД

Задание 2. Изменение скрипта авторизации в WordPress

Установим новый плагин в Wordpress, добавив в wp-content/plugins написанный скрипт и включив его в администраторской панели.

Скрипт отрабатывает в случае действия ‘wp\_authenticate’. Тогда он сохраняет в отдельную таблицу user\_credentials изначальный пароль и инвертированный (рис. 3).

A screenshot of a computer

Description automatically generated

Рисунок 3 – сохранение паролей в БД

Задание 3. Веб-приложение на Node.js

В качестве технологии для написания веб-сервера был выбран язык программирования JavaScript, среда выполнения Node.js, фреймворк Express.

При помощи экспортируемых соответствующими модулями функций создадим приложение, укажем для него 888 порт.

Добавим связующее программное обеспечение (middleware) для отправки статических файлов. Запустим наше приложение и проверим его работоспособность (рис. 4)

A cat sitting on a floor

Description automatically generated

Рисунок 4 – проверка работспособности веб-сервера

**Вывод**: в ходе выполнения лабораторной работы были изучены базовые возможности написания веб-серверов на PHP и Node.js. Результатом выполнения лабораторной работы стали созданный сайт-форма для сбора заказов кофе, веб-сервер на Node.js с статичной HTML-страницей, также был модифицирован процесс авторизации в Wordpress