ФГБОУ ВО «СЕВЕРО-ОСЕТИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ К.Л. ХЕТАГУРОВА»

Факультет математики и компьютерных наук

Кафедра прикладной математики и информатики

ОТЧЕТ ПО ПРАКТИКЕ

Наименование практики: Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы))

направление подготовки 01.03.02 Прикладная математика и информатика,

профиль «Программирование, анализ данных и математическое моделирование»

Выполнил студент Баскаев Таймураз Олегович \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

*(ФИО) (подпись) (оценка)*

Дата сдачи отчета: «26» июля 2025 г.

Отчет принят: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.К.Гутнова / «26» июля 2025 г..

(подпись) *Ф.И.О.* преподавателя-экзаменатора *(дата)*

Содержание

Введение 3

Рекомендационная система для гостиничного бизнеса 6

Заключение 15

Список литературы 16

Приложение 1 17

Приложение 2 20

Приложение 3 22

Введение

**Целями** Учебной практики (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)) (далее – Практика) является формирование представлений о специфике научно-исследовательской деятельности, осуществляемой при написании выпускной квалификационной и курсовых работ; а также формирование исследовательских качеств и умений, необходимых в профессиональной деятельности.

**Задачи Практики**:

– ознакомление с различными этапами научно-исследовательской деятельности (постановка задачи исследования, литературная проработка проблемы с использованием современных информационных технологий, накопление и анализ экспериментального (теоретического) материала, формулировка выводов по итогам исследований, оформление результатов работы в виде отчета);

– ознакомление с различными методами научного поиска, выбор оптимальных методов исследования, соответствующих задачам исследования;

– приобретение навыков коллективной научно-исследовательской деятельности, взаимодействие с другими научными группами и исследователями;

– формирование умений находить, анализировать и систематизировать научную, справочную, статистическую информацию, в том числе с использованием современной компьютерной техники и ИКТ;

– формирование навыка применения результатов исследования в профессиональной деятельности.

**Место прохождения практики**: кафедра прикладной математики и информатики.

**Форма проведения Практики**: непрерывно.

**Способ проведения Практики**: стационарная.

**Период прохождения практики**: с 30.06.2025 по 26.07.2025 включительно (4 недели).

Руководство практикой осуществлялось доцентом кафедры прикладной математики и информатики Гутновой А.К.

Прохождение практики осуществлялось в соответствии с рабочим графиком (см. приложение 1) и индивидуальным заданием (см. приложение 2).

**Краткое содержание и цель индивидуального задания**:

Целью индивидуального задания являлась разработка системы рекомендаций отелей на основе контентной фильтрации (Content-Based Filtering). Проект включал в себя три основных этапа: парсинг данных с сайта Booking.com, разработку бэкенда с логикой рекомендаций и создание простого фронтенда для отображения результатов.

**Перечень выполненных работ и заданий**. В соответствии с индивидуальным заданием за период прохождения практики выполнена следующая работа:

На первом этапе осуществлялся парсинг данных с сайта Booking.com. Для этого был разработан скрипт на языке Python с использованием библиотек Selenium и BeautifulSoup. Скрипт автоматизировал сбор ключевой информации об отелях, включая их названия, адреса, описания, отзывы, удобства и другие параметры. Полученные данные сохранялись в структурированном формате JSON для последующей обработки. Далее велась разработка бэкенд-части проекта. Было создано FastAPI-приложение, обеспечивающее обработку запросов и взаимодействие с фронтендом. Основной задачей этого этапа стала реализация рекомендательной системы, основанной на контентной фильтрации (Content-Based Filtering). Для этого применялась модель Sentence Transformers, которая преобразовывала текстовые данные в векторные представления, а библиотека FAISS использовалась для эффективного поиска похожих отелей. На заключительном этапе разрабатывался простой фронтенд. Были созданы HTML-страницы, отображающие список отелей и детальную информацию о каждом из них. Для динамического взаимодействия с бэкендом использовался JavaScript, а также применялся CSS для стилизации интерфейса и улучшения пользовательского опыта. Таким образом, в ходе практики была успешно реализована система рекомендаций отелей, включающая сбор данных, их обработку и визуализацию результатов. Все виды деятельности в период прохождения практики отражены в Дневнике практики (см. приложение 3).

Рекомендационная система для гостиничного бизнеса

1. Постановка задачи

Проект представляет собой рекомендательную систему для гостиничного бизнеса, которая помогает пользователям находить отели, похожие на те, которые они уже рассматривают. Система использует данные с сайта Booking.com, включая описания отелей, отзывы, удобства и расположение. Основная цель — предоставить персонализированные рекомендации на основе контента, что улучшает пользовательский опыт и упрощает поиск подходящих вариантов.

2. Руководство администратора

2.1 Подготовка серверной среды.

Перед запуском системы необходимо убедиться, что на сервере установлен Python версии 3.8 или выше. Для работы проекта требуются дополнительные зависимости, которые устанавливаются через файл requirements.txt. Для этого выполняется команда:

pip install -r requirements.txt

2.2 Запуск бэкенда

Бэкенд системы работает на FastAPI и использует uvicorn в качестве ASGI сервера. Для запуска сервера необходимо перейти в директорию с файлом main.py и выполнить команду:

uvicorn main:app --host 0.0.0.0 --port 8000

Сервер запустится на порту 8000 и будет доступен по адресу [http://localhost:8000](http://localhost:8000/). Если требуется работа в фоновом режиме (например, на удаленном сервере), можно использовать screen, tmux или systemd для управления процессом.

2.3 Работа с данными

Система использует заранее подготовленные данные отелей в формате JSON. Файл hotels\_data.json должен находиться в корневой директории проекта (в папках backend и frontend). Если данных нет, их необходимо получить, запустив парсер из файла SILENIUMVER2GPT.PY. Для этого выполняется команда:

python SILENIUMVER2GPT.PY

Парсер соберет актуальную информацию с Booking.com и сохранит её в hotels\_data.json. После этого система автоматически обработает данные при первом запросе рекомендаций, создав файлы clean\_hotels.json, hotel\_vectors.npy и hotel\_index.faiss.

3. Руководство программиста

3.1. Общая архитектура системы

Проект представляет собой рекомендательную систему для отелей, работающую по принципу content-based filtering. Система состоит из трех ключевых компонентов: парсер, бэкенд, фронтенд. Парсер (Selenium + BeautifulSoup) – собирает данные с Booking.com и сохраняет их в JSON. Бэкенд (FastAPI + Sentence Transformers + FAISS) – обрабатывает данные, строит векторные представления отелей и выдает рекомендации. Фронтенд (HTML/CSS/JS) – отображает отели и рекомендации, взаимодействуя с API. Система использует эмбеддинги текста (модель paraphrase-MiniLM-L6-v2) и метрический поиск (FAISS) для быстрого нахождения похожих отелей.

3.2. Работа с парсером (SILENIUMVER2GPT.PY)

Парсер собирает данные с Booking.com и сохраняет их в hotels\_data.json.

Основные функции парсера:

get\_driver() – настраивает Selenium WebDriver для работы в headless-режиме, download\_hotel\_page(url) – загружает HTML-страницу отеля с динамическим контентом, parse\_all\_hotel\_data(html) – извлекает данные из HTML (название, адрес, описание, отзывы, удобства и др.), get\_hotel\_links(search\_url) – находит ссылки на отели в результатах поиска Booking.com, scrape\_and\_save\_hotels(search\_url) – основной цикл парсинга.

3.3. Бэкенд (main.py и recommender.py)

**FastAPI-сервер (main.py)**

/recommend (POST) – принимает данные отеля, возвращает рекомендации.

/hotels\_data.json (GET) – отдает JSON с отелями для фронтенда.

Статический фронтенд – монтируется через StaticFiles.

**Рекомендательная система (recommender.py)**

hotels\_data.json → clean\_hotels.json (очищенные данные). Если файлов нет – система парсит и обрабатывает данные автоматически.

**Векторизация отелей (hotel\_to\_vector)**

Процесс векторизации отелей начинается с преобразования текстовой информации в числовые векторы. Система объединяет описание отеля, его адрес и перечень удобств в единый текстовый блок. Этот блок обрабатывается специальной нейросетевой моделью (sentence-transformers), которая преобразует текст в компактный числовой вектор - эмбеддинг размером 384 значения. Параллельно извлекаются оценки по различным критериям (персонал, чистота, удобства и другие), которые формируют дополнительный числовой вектор. Эти два вектора объединяются в один общий вектор, который и представляет отель в математическом пространстве.

Поиск рекомендаций осуществляется с помощью оптимизированной системы FAISS. Все векторы отелей заранее сохраняются в специальной индексированной структуре (IndexFlatL2), что позволяет быстро находить похожие варианты. Когда пользователь выбирает конкретный отель, система вычисляет расстояния между его вектором и векторами всех других отелей в базе данных. В результате возвращаются 5 отелей с наиболее близкими векторами, которые и предлагаются как рекомендации. Этот подход обеспечивает быстрый и точный поиск даже среди тысяч вариантов.

3.4. Фронтенд (index.html, hotel.html, script.js, style.css)

**Главная страница (index.html)**

Отображает карточки отелей из hotels\_data.json.

При клике сохраняет выбранный отель в localStorage и переходит на hotel.html.

**Страница отеля (hotel.html)**

Показывает детали отеля и похожие отели (через API /recommend).

**API-запросы (script.js)**

fetch("hotels\_data.json") – загружает список отелей.

fetch("/recommend", { method: "POST" }) – получает рекомендации.

4. Гиперпараметры

В рекомендательной системе применяется набор ключевых параметров, определяющих эффективность работы алгоритмов. Эти настройки можно условно разделить на несколько категорий, каждая из которых влияет на различные аспекты функционирования системы.

**Модель формирования векторных представлений** использует предобученный трансформер paraphrase-MiniLM-L6-v2, который создает эмбеддинги размерностью 384 компонента. Эта модель была выбрана как оптимальный баланс между качеством представления текстовой информации и вычислительной эффективностью. Альтернативным вариантом могла бы стать модель all-MiniLM-L12-v2 с размерностью 512 компонентов, что потенциально улучшило бы качество рекомендаций, но потребовало бы больше ресурсов для обработки.

**Параметры поиска похожих объектов** включают несколько важных настроек. Для поиска ближайших соседей используется точный алгоритм IndexFlatL2 из библиотеки FAISS, который вычисляет евклидово расстояние между векторами. Количество возвращаемых рекомендаций по умолчанию установлено равным пяти. Это значение можно регулировать в зависимости от требований к системе - увеличение числа рекомендаций даст пользователю больше вариантов для выбора, но может снизить релевантность результатов.

**Обработка входных данных** предусматривает особый подход к формированию признакового пространства. Текстовые данные, включающие описание отеля, его адрес и перечень удобств, объединяются в единую строку перед векторизацией. Оценочные характеристики из отзывов, такие как оценки персонала, чистоты и других параметров, преобразуются в числовой вектор фиксированной размерности. В текущей реализации эти два типа признаков объединяются без явного взвешивания, что оставляет возможность для дальнейшей оптимизации.

**Настройки парсинга данных** содержат параметры, обеспечивающие стабильную работу сбора информации с сайта-источника. Временные задержки между запросами установлены в диапазоне от 1.2 до 2.5 секунд, что позволяет избежать блокировки со стороны сервера. Алгоритм автоматической прокрутки страницы ограничен десятью попытками, что обеспечивает баланс между полнотой сбора данных и временем выполнения.

5. Методы и алгоритмы решения

Основу системы составляет **функция**hotel\_to\_vector(), которая преобразует данные отеля в числовой вектор. Вот как это работает:

def hotel\_to\_vector(hotel):

# Объединяем текстовые данные

description = hotel.get("description", "")

address = hotel.get("address", "")

amenities = hotel.get("amenities\_text", "")

text = f"{description} {address} {amenities}"

# Получаем текстовый эмбеддинг

text\_embedding = model.encode([text])[0]

# Формируем вектор из оценок

reviews = hotel.get("reviews", {})

review\_vector = np.array([reviews.get(k, 0) for k in review\_keys])

# Объединяем оба вектора

return np.concatenate([text\_embedding, review\_vector]).astype("float32")

Эта функция сначала создает единую текстовую строку из описания, адреса и удобств, затем преобразует ее в 384-мерный вектор с помощью предобученной модели. Отдельно формируется вектор из оценок по ключевым категориям. Финальный вектор получается объединением этих двух компонент.

5.1. Устройство системы рекомендаций

Ядро системы работает через **функцию**recommend\_similar():

def recommend\_similar(input\_hotel, top\_k=5):

# Преобразуем входной отель в вектор

input\_vec = hotel\_to\_vector(input\_hotel).reshape(1, -1)

# Ищем ближайшие векторы в индексе FAISS

distances, indices = index.search(input\_vec, top\_k)

# Возвращаем соответствующие отели

return [hotels\_data[i] for i in indices[0]]

Здесь входной отель преобразуется в вектор, FAISS-индекс находит ближайшие векторы, возвращаются реальные данные отелей

Индекс строится при первом запуске системы:

# Создание FAISS индекса

dim = vectors.shape[1]

index = faiss.IndexFlatL2(dim)

index.add(vectors)

Система использует **гибридный подход**:

* Для текстовой части применяется **трансформерная модель** (sentence-transformers), которая понимает семантику текста
* Для числовых оценок используется прямое включение в вектор
* Поиск соседей выполняется через **оптимизированный алгоритм FAISS**

Особенность реализации в том, что все данные **предварительно индексируются**:

# При первом запуске системы

hotels\_data = load\_hotels()

vectors = np.array([hotel\_to\_vector(h) for h in hotels\_data], dtype="float32")

index = faiss.IndexFlatL2(vectors.shape[1])

index.add(vectors)

Это позволяет затем быстро (за миллисекунды) находить похожие отели, даже если в базе тысячи вариантов. Алгоритм использует **евклидово расстояние** между векторами как меру похожести, что хорошо работает для такого типа данных.

Ключевая оптимизация - **разделение процессов**:

1. Тяжелая операция векторизации выполняется один раз при загрузке данных
2. Поиск работает с уже подготовленными векторами
3. Индекс FAISS оптимизирован для быстрого поиска ближайших соседей

Такая архитектура обеспечивает мгновенный отклик системы при запросе рекомендаций, что критически важно для пользовательского опыта.

6. Этапы выполненных работ

Работа над проектом велась поэтапно, начиная с разработки надежного парсера для сбора данных. Специально написанный скрипт на Python успешно обрабатывал страницы Booking.com, извлекая всю необходимую информацию и сохраняя ее в структурированном виде.

На следующем этапе было создано FastAPI-приложение, включающее эндпоинт для обработки запросов на рекомендации. Особое внимание уделялось оптимизации алгоритма векторизации текстовых данных и поиска похожих отелей, чтобы обеспечить быстрый отклик системы даже при работе с большими объемами данных.

Фронтенд часть проекта включает две основные страницы: главную с перечнем отелей и страницу с детальной информацией. Интерфейс разработан с учетом принципов юзабилити, что делает взаимодействие с системой интуитивно понятным для конечных пользователей.

7. Результаты решения задачи

В результате проведенной работы успешно создана полнофункциональная система рекомендаций отелей. Парсер демонстрирует стабильную работу и способен собирать актуальные данные с Booking.com. Алгоритм рекомендаций корректно анализирует текстовые описания и подбирает схожие отели, что подтверждено тестовыми запусками системы.

Разработанный веб-интерфейс обеспечивает удобный доступ к функциональности системы. Пользователи могут просматривать список отелей, изучать их подробные характеристики и получать персонализированные рекомендации.

Перспективы развития проекта включают расширение функциональности за счет внедрения механизма учета пользовательских предпочтений, улучшение адаптивности интерфейса для мобильных устройств.

Ссылка на git репозиторий проекта:

<https://github.com/baskaev/ContentBasedRecomendationSystemHotelsAndParsingBooking.git>

Заключение

Работая над проектом, я разобрался, как устроены content-based рекомендательные системы. Теперь понимаю, как преобразовывать описания отелей и другие данные в векторы, чтобы находить похожие варианты. Основное время ушло на подбор правильного способа комбинирования текстовой информации и числовых оценок.

С библиотекой FAISS получилось не сразу - пришлось повозиться с настройками, чтобы найти баланс между скоростью работы и качеством рекомендаций. Особенно сложно было подобрать оптимальные параметры для индексации. На практике увидел, как сильно результаты зависят от качества исходных данных - пришлось несколько раз переделывать парсер, чтобы получить более чистые данные.

Главное, что я вынес - рекомендательные системы требуют постоянной тонкой настройки. То, что работает в теории, на практике часто дает неожиданные результаты. Например, некоторые отели оказывались "похожими" только потому, что у них были одинаковые оценки по какому-то неочевидному параметру.

Полученный опыт пригодится мне в будущем, особенно понимание всего цикла - от сбора данных до итоговых рекомендаций. Хотя система работает, я вижу несколько направлений для улучшения, в частности - добавление дополнительных параметров для сравнения отелей.

Список литературы

1. Aggarwal C. C. Recommender Systems: The Textbook. — Cham: Springer, 2016. — 498 p.
2. Грузман А. Я. Разработка рекомендательных систем: учебное пособие. — Москва: ДМК Пресс, 2021. — 234 с.
3. Beautiful Soup Documentation [Электронный ресурс]. — URL: [https://www.crummy.com/software/BeautifulSoup/bs4/doc/](https://vk.com/away.php?to=https%3A%2F%2Fwww.crummy.com%2Fsoftware%2FBeautifulSoup%2Fbs4%2Fdoc%2F&utf=1) (дата обращения: 15.07.2025).
4. FAISS Documentation [Электронный ресурс]. — URL: [https://github.com/facebookresearch/faiss](https://vk.com/away.php?to=https%3A%2F%2Fgithub.com%2Ffacebookresearch%2Ffaiss&utf=1) (дата обращения: 15.07.2025).
5. Selenium Python Documentation [Электронный ресурс]. — URL: [https://selenium-python.readthedocs.io](https://vk.com/away.php?to=https%3A%2F%2Fselenium-python.readthedocs.io&utf=1) (дата обращения: 15.07.2025).

**Приложение 1**

ФГБОУ ВО «Северо-Осетинский государственный университет

имени Коста Левановича Хетагурова»

Кафедра прикладной математики и информатики

УТВЕРЖДАЮ

заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_ / Е.К. Басаева /

«30» июня 2025 г.

|  |  |  |
| --- | --- | --- |
|  | | |
| **РАБОЧИЙ ГРАФИК (ПЛАН) ПРОВЕДЕНИЯ ПРАКТИКИ** | | |
| **Общие сведения** | | |
| ФИО обучающегося | Баскаев Таймураз Олегович |
| Курс, группа, форма обучения | *3 курс, ПМ(б)-22-1-ОФО* |
| Направление подготовки, профиль | *01.03.02 Прикладная математика и информатика,*  *профиль «Программирование, анализ данных и математическое моделирование»* |
| Наименование структурного подразделения (кафедра) | *кафедра прикладной математики и информатики* |
| Вид практики | *учебная практика* |
| Тип практики | *научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)* |
| Способ проведения практики | *стационарная* |
| Форма проведения практики | *непрерывно* |
| Место прохождения практики | *кафедра прикладной математики и информатики* |
| Период прохождения практики | *с «30» июня 2025 г. по «26» июля 2025 г.* |
| Реквизиты договора о прохождении практики (при проведении практики в профильной организации) | – |

**Планируемые работы**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Содержание работы | Срок выполнения | Отметка о выполнении |
|  | Оформление документов по прохождению практики | до начала практики | не требуется |
|  | Проведение медицинских осмотров (обследований) в случае выполнения обучающимся работ, при выполнении которых проводятся обязательные предварительные и периодические медицинские осмотры (обследования) в соответствии с законодательством РФ | до начала практики | не требуется |
|  | Установочный инструктаж по целям, задачам, срокам и требуемой отчетности. | *30.06.2025 1230–1400* | выполнено |
|  | Вводный инструктаж по правилам охраны труда, технике безопасности, пожарной безопасности, оформление временных пропусков для прохода в профильную организацию (при необходимости). | *30.06.2025* | выполнено |
|  | Оформление индивидуального задания и плана работы. | *01.07.2025 1040–1300* | выполнено |
|  | Выполнение индивидуального задания практики | *30.06.25-04.07.25*  *07.07.25-11.07.25*  *14.07.25-18.07.25*  *9.00-12.00* | выполнено |
|  | Консультации руководителя(-ей) практики о ходе выполнения заданий, оформлении и содержании отчета, по производственным вопросам | *30.06.25-04.07.25*  *07.07.25-11.07.25*  *14.07.25-18.07.25*  *9.00-13.00* | выполнено |
|  | Подготовка отчета по практике | *21.07.2025–25.07.2025* | выполнено |
|  | Проверка отчета по практике, оформление характеристики руководителя(-ей) практики | *26.07.2025 900–1300* | выполнено |
|  | Промежуточная аттестация по практике | *в соответствии с расписанием сессии* |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Рабочий график (план) составил: | | | | | | | | | | | | |
|  | руководитель практики от образовательной организации | | | | | | | | | | | | |
|  | к.ф.-м.н., доцент |  | |  | |  | | Гутнова А.К. | | |  | | «30» июня 2025 г |
|  | (уч. степень, уч. звание, должность) |  | | (подпись) | |  | | (И.О. Фамилия) | | |  | | (дата) |
|  |  | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | |  | |  | |  | |  |  | |  | |
|  | С рабочим графиком (планом) ознакомлен: | | | | | | | | | | | | |
|  | обучающийся |  | |  | |  | | Баскаев Т.О. | | |  | | «30» июня 2025 г |
|  |  |  | | (подпись) | |  | | (И.О. Фамилия) | | |  | | (дата) |

**Приложение 2**

ФГБОУ ВО «Северо-Осетинский государственный университет

имени Коста Левановича Хетагурова»

Кафедра прикладной математики и информатики

УТВЕРЖДАЮ

заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_ / Е.К. Басаева /

«30» июня 2025 г.

|  |  |
| --- | --- |
|  | |
| **ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ** **НА ПРАКТИКУ** | |
| **Общие сведения** | |
| ФИО обучающегося | Баскаев Таймураз Олегович |
| Курс, группа, форма обучения | *3 курс, ПМ(б)-22-1-ОФО* |
| Направление подготовки, профиль | *01.03.02 Прикладная математика и информатика,*  *профиль «Программирование, анализ данных и математическое моделирование»* |
| Наименование структурного подразделения (кафедра / отделение) | *кафедра прикладной математики и информатики* |
| Вид практики | *учебная практика* |
| Тип практики | *научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)* |
| Способ проведения практики | *стационарная* |
| Форма проведения практики | *непрерывно* |
| Место прохождения практики | *кафедра прикладной математики и информатики* |
| Период прохождения практики | *с «30» июня 2025 г. по «26» июля 2025 г.* |
| Реквизиты договора о прохождении практики (при проведении практики в профильной организации) | – |
|  | |
| **Содержание индивидуального задания**  Разработать систему рекомендаций отелей на основе контентной фильтрации. Преобразовать данные в векторные представления через Sentence Transformers. Реализовать поиск похожих отелей с помощью FAISS. Настроить параметры сравнения. Сделать бэкенд на FastAPI с API для рекомендаций. Сверстать фронтенд (HTML/CSS/JS) для отображения отелей и рекомендаций. Организовать связку фронта и бэка. | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Задание на практику составил:  руководитель практики от образовательной организации | | | | | | | |
|  | к.ф.-м.н., доцент |  |  |  | Гутнова А.К. |  | «30» июня 2025 г | |
|  | (уч. степень, уч. звание, должность) |  | (подпись) |  | (И.О. Фамилия) |  | (дата) | |
|  |  | | | | | | | |
|  | | | | | | | |
|  |  | | | | | | | |
|  | Задание на практику принял: | | | | | | | |
|  | обучающийся |  |  |  | Баскаев Т.О. |  | «30» июня 2025 г | |
|  |  |  | (подпись) |  | (И.О. Фамилия) |  | (дата) | |

**Приложение 3**

ФГБОУ ВО «Северо-Осетинский государственный университет

имени Коста Левановича Хетагурова»

Кафедра прикладной математики и информатики

**ДНЕВНИК ПРАКТИКИ**

**Общие сведения**

|  |  |
| --- | --- |
| ФИО обучающегося | Баскаев Таймураз Олегович |
| Курс, группа, форма обучения | *3 курс, ПМ(б)-22-1-ОФО* |
| Направление подготовки, профиль | *01.03.02 Прикладная математика и информатика,*  *профиль «Программирование, анализ данных и математическое моделирование»* |
| Наименование структурного подразделения (кафедра / отделение) | *кафедра прикладной математики и информатики* |
| Вид практики | *учебная практика* |
| Тип практики | *научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)* |
| Способ проведения практики | *стационарная* |
| Форма проведения практики | *непрерывно* |
| Место прохождения практики | *кафедра прикладной математики и информатики* |
| Период прохождения практики | *с «30» июня 2025 г. по «26» июля 2025 г.* |
| Реквизиты договора о прохождении практики (при проведении практики в профильной организации) | – |

**Учет выполняемой работы**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Содержание работы | Дата выполнения | Отметка о выполнении |
|  | Выдача и оформление индивидуального плана прохождения практики и индивидуального задания на практику | 30.06.2025 | выполнено |
|  | Парсинг данных с Booking.com, сохранение в JSON | 30.06.2025–04.07.2025 | выполнено |
|  | Разработка бэкенда и логики рекомендаций | 07.07.2025–11.07.2025 | выполнено |
|  | Создание фронтенда и интеграция с бэкендом | 14.07.2025–18.07.2025 | выполнено |
|  | Подготовка отчета по практике.  Защита отчета по практике. | 21.07.2025–26.07.2025 | выполнено |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Дневник заполнил: | | | | | | |
| обучающийся |  |  |  | Баскаев Т.О. |  | 26 июля 2025 г. |
|  |  | (подпись) |  | (И.О. Фамилия) |  | (дата) |
|  | | | | | | |
| Дневник проверил: | | | | | | |
| руководитель практики от образовательной организации | | | | | | |
| к.ф.-м.н., доцент |  |  |  | Гутнова А.К. |  | 26 июля 2025 г. |
| (уч. степень, уч. звание, должность) |  | (подпись) |  | (И.О. Фамилия) |  | (дата) |