|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ**

Студент *Ким Алексей Максимович*

*фамилия, имя, отчество*

Группа *ИУ5-82б*

Название предприятия *НУК ИУ МГТУ им. Н.Э.Баумана*

Научный руководитель *29.05.2024*  ***Максаков А.А.***

*дата, подпись фамилия, и.о.*

Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Руководитель практики от кафедры *29.05.2024*  ***Кротов Ю.Н.***

*дата, подпись фамилия, и.о.*

Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Москва 2024г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра «Системы обработки информации и управления»

**ЗАДАНИЕ**

**на прохождение преддипломной практики**

на предприятии \_\_\_\_МГТУ им. Н.Э.Баумана

Студент\_\_*Ким Алексей Максимович*\_\_\_ Группа \_\_\_\_\_ИУ5-82б\_\_\_\_\_\_\_\_

(фамилия, имя, отчество; инициалы; индекс группы)

Во время прохождения производственной практики студент **должен**:

1. Подобрать программные средства для работы системы прогнозирования загрузки СХД.
2. Составить список функций, компонентов системы и используемых технологии.
3. Разработать программную часть системы прогнозирования загрузки СХД.
4. Подготовить и сдать отчет о практике до 30.05.2024 года.

Дата выдачи задания « \_15\_ » \_мая\_\_ 2024 г.

Студент **\_\_***15.05.2024***\_\_\_\_\_\_ *Ким А.М.\_\_\_***

*подпись, дата фамилия, и.о.*

Научный руководитель **\_\_***15.05.2024***\_\_\_\_\_\_ \_\_\_*Максаков А.А.*\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики от кафедры **\_\_***15.05.2024***\_\_\_\_\_\_ \_\_\_*Кротов Ю.Н.*\_\_\_**

*подпись, дата фамилия, и.о.*

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc167822523)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc167822524)

[1 Требования к системе 5](#_Toc167822525)

[1.1 Требования к аппаратным средствам 5](#_Toc167822526)

[1.2 Требования к программным средствам 5](#_Toc167822527)

[2 Определение функционала системы 5](#_Toc167822528)

[3 Анализ используемых технологий 6](#_Toc167822529)

[3.1. Анализ фреймворков для создания интерфейса для работы с 6](#_Toc167822530)

[пользователем 6](#_Toc167822531)

[3.2. Анализ фреймворков для бэкенда 8](#_Toc167822532)

[ЗАКЛЮЧЕНИЕ 11](#_Toc167822533)

[ПРИЛОЖЕНИЕ A 12](#_Toc167822534)

ВВЕДЕНИЕ

Актуальность создания системы обусловлена стремительным ростом объемов данных в современном мире. С каждым годом количество генерируемой информации увеличивается, что связано с рядом факторов. Во-первых, увеличение числа интернет-пользователей в мире способствует генерации большего объема данных. Во-вторых, рост популярности контента высокого качества, требует больших мощностей для их хранения и обработки.

Системы хранения данных (СХД) играют ключевую роль в управлении этими объемами информации. Они не только обеспечивают хранение данных, но и гарантируют их доступность и безопасность, что является критически важным для бизнес-процессов современных компаний. Перегрузка таких систем может привести к серьезным сбоям, вплоть до полной потери данных, что недопустимо в бизнес-среде.

В связи с этим возникает острая необходимость в прогнозировании загрузки на системы хранения данных. Эффективное прогнозирование позволяет не только оптимизировать использование ресурсов, но и предотвратить длительные простои, минимизируя тем самым потери данных и финансовые издержки. Для решения этой задачи активно используются современные методы искусственного интеллекта и машинного обучения, которые позволяют анализировать большие объемы информации и делать точные прогнозы в реальном времени.

ОСНОВНАЯ ЧАСТЬ

1. **Требования к системе**

Работа системы должна покрывать следующие задачи:

* предоставлять пользователю возможность гибко настраивать систему;
* прогнозировать загрузку СХД;
* рисовать прогноз на графике;
* гибко настраивать график;
* предлагать пользователям интуитивно понятный интерфейс;
* предоставлять пользователю информацию о возникающих ошибках;
* предоставить пользователю возможность работы с разных устройств.
  1. **Требования к аппаратным средствам**

Операционная система, которая поддерживает интерпретатор Python , NodeJS и postrgres: Linux, macOS, или Windows. Интерпретатор языка программирования Python: Python 3.10. Установленные библиотеки numpy, pandas, tenacity, requests, matplotlib, plotly, scipy, scikit-learn, fastapi; также нужен NodeJS с установленным фреймворком React и библиотеками: plotly, axios, styled-component, react-spinners; и БД postgres.

* 1. **Требования к программным средствам**

Не предъявляются.

1. **Определение функционала системы**

Для определения функционала системы было проведена встреча с заказчиком системы:

* выбор кластера СХД,
* возможность выбрать параметры
* возможность изменения уровня прогнозирования,
* настройка скользящего окна,
* отображение облака точек.

1. **Анализ используемых технологий**

Определение технологий, на основе которых будет разрабатываться система – важный этап проектирования. Благодаря грамотно подобранным технологиям, можно существенно ускорить процесс разработки.

## Анализ фреймворков для создания интерфейса для работы с

## пользователем

Интерфейс является важной частью системы, поскольку без него невозможно взаимодействие пользователя с программой, следовательно, теряется смысл существования программы.

Создание интерфейса, с которым будет работать пользователь также является очень трудоемкой задачей, поэтому важно правильно выбрать механизм его разработки.

Выбор фреймворка для разработки веб-приложений – важное решение, которое может существенно повлиять на эффективность и качество конечного продукта. React, Angular и Vue – три самых популярных фреймворка, каждый из которых имеет свои сильные и слабые стороны. Давайте рассмотрим их подробнее и определим, почему React может быть лучшим выбором для вашего проекта.

Таблица 1 – Сравнение React, Angular и Vue.

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр сравнения | **React** | **Angular** | **Vue** |
| Популярность и сообщество | Очень популярный, большое сообщество | Популярный, поддерживается Google | Популярный, быстро растущее сообщество |
| Сложность обучения | Средняя, требуется знание JavaScript и JSX | Высокая, требует знания TypeScript и Angular-specific синтаксиса | Низкая, простой для старта |
| Архитектура | Библиотека для создания UI, гибкость | Полный фреймворк MVC | Прогрессивный фреймворк, может использоваться как библиотека |
| Производительность | Высокая | Высокая | Высокая |
| Инструменты и экосистема | Развитая экосистема, много инструментов | Полный набор инструментов из коробки | Развитая экосистема, но меньше чем у React |
| Примеры использования | Веб-приложения, мобильные приложения | Корпоративные приложения, сложные SPA | Прогрессивные веб-приложения, интерфейсы |

Вывод: React выделяется среди других фреймворков своей гибкостью и универсальностью, предоставляя разработчикам свободу в выборе архитектуры и интеграции с другими инструментами. Его популярность и большое сообщество делают его привлекательным благодаря доступности множества решений, библиотек и обучающих материалов. Использование виртуального DOM обеспечивает высокую производительность, что важно для сложных приложений. Богатая экосистема инструментов, таких как React Router и Redux, упрощает разработку и управление состоянием. Регулярные обновления и активная поддержка со стороны Facebook гарантируют долгосрочную стабильность и актуальность. Эти качества делают React оптимальным выбором для создания современных веб-приложений.

## Анализ фреймворков для бэкенда

Выбор подходящего фреймворка для разработки бэкенда – это важный шаг, который может значительно повлиять на производительность и масштабируемость вашего приложения. Django, Flask и FastAPI – три популярных фреймворка на Python, каждый из которых имеет свои особенности и преимущества. Давайте рассмотрим их подробнее и определим, какой из них лучше подходит для различных задач.

Таблица 1 – Сравнение Django, Flask и FastAPI.

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Django | Flask | FastAPI |
| Популярность и сообщество | Очень популярный, большое сообщество | Популярный, активное сообщество | Быстро растущая популярность, поддержка от крупных компаний |
| Простота настройки и конфигурации | Средняя, много встроенных возможностей, требуется конфигурация | Высокая, минимальные настройки изначально | Высокая, минимальные настройки изначально |
| Поддержка баз данных | Встроенный ORM, поддержка множества СУБД | Нет встроенного ORM, поддержка через сторонние библиотеки | Поддержка через сторонние библиотеки, включая SQLAlchemy и Tortoise-ORM |
| Модульность и расширяемость | Высокая, множество встроенных модулей | Очень высокая, легко расширяем через плагины | Высокая, легко интегрируется с внешними библиотеками и плагинами |
| Безопасность | Встроенные механизмы безопасности, включая защиту от CSRF и XSS | Основные механизмы безопасности, требуется настройка | Встроенные механизмы безопасности, включая OAuth2 и JWT |
| Асинхронное программирование | Ограниченная поддержка | Поддержка через расширения | Изначально разработан для асинхронного программирования, отличная поддержка |
| Документация и обучающие ресурсы | Отличная, много обучающих ресурсов | Отличная, много примеров и документации | Отличная, подробная документация и примеры |
| Обновления и активность разработки | Регулярные обновления, активное развитие | Регулярные обновления, активное сообщество | Частые обновления, активное развитие и поддержка от крупных компаний |
| Поддержка развертывания и DevOps | Поддержка через Django management commands, интеграция с CI/CD | Легкая интеграция с CI/CD, поддержка развертывания через Fabric и другие инструменты | Хорошая поддержка CI/CD, инструменты для автоматического развертывания |
| Соответствие различным видам проектов | Подходит для крупных веб-приложений, корпоративных проектов | Идеально для микросервисов, небольших приложений и API | Отлично подходит для высоконагруженных API, микросервисов и современных приложений |

Django отлично подходит для создания крупных веб-приложений и корпоративных проектов благодаря своей полной архитектуре и множеству встроенных возможностей. Встроенные механизмы безопасности, поддержка множества баз данных и отличная документация делают его надежным выбором для серьезных проектов.

Flask обеспечивает максимальную гибкость и простоту настройки, что делает его идеальным для микросервисов и небольших приложений. Легко расширяемый и интуитивно понятный, Flask предоставляет разработчикам свободу в выборе инструментов и библиотек.

FastAPI выделяется своей производительностью и поддержкой асинхронного программирования, что делает его идеальным для создания высоконагруженных API и современных приложений. Быстрая настройка, встроенные механизмы безопасности и активное развитие делают FastAPI привлекательным выбором для разработчиков, стремящихся использовать современные технологии и подходы.

В итоге, выбор фреймворка зависит от специфики вашего проекта. Django подойдет для крупных и сложных проектов, Flask обеспечит гибкость и простоту для небольших приложений, а FastAPI предоставит высокую производительность и современные возможности для создания эффективных API.

ЗАКЛЮЧЕНИЕ

В ходе выполнения преддипломной практической работы были выполнены следующие задачи:

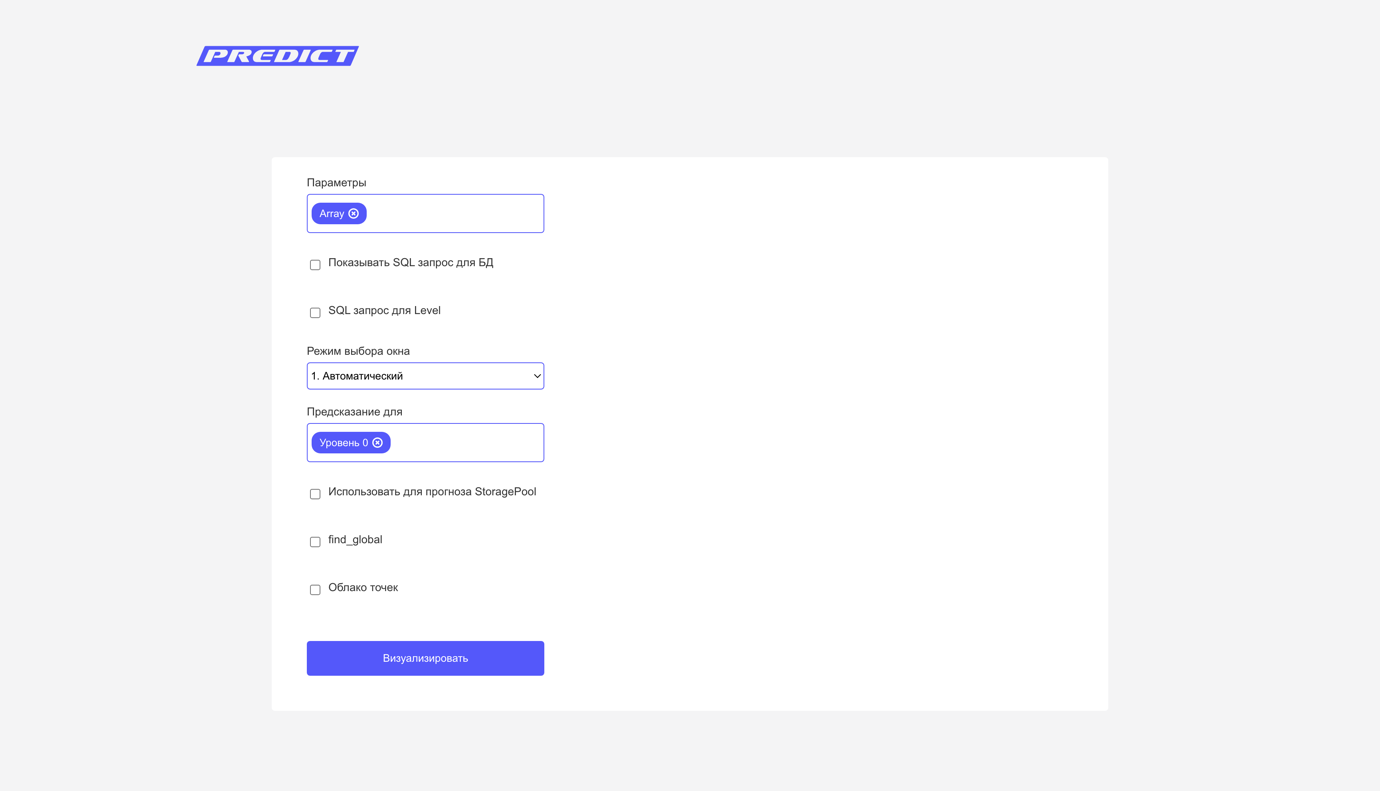
1. Подобраны аппаратные и программные средства для системы прогнозирования нагрузки СХД.
2. Проведен анализ инструментов создания системы.
3. Спроектирована программная часть.
4. Разработана программная часть системы.

ПРИЛОЖЕНИЕ A

**ГРАФИЧЕСКИЕ МАТЕРИАЛЫ**

1. Начальный интерфейс;
2. Выбор параметров;
3. Чекбокс показать SQL запрос для БД;
4. Чекбокс запрос для Level;
5. Выбор режима окна;
6. Выбор уровня прогнозирования;
7. Чекбоксы – использовать для прогноза StoragePool, find\_global и облако точек.
8. Интерфейс после нажатия кнопки визуализировать

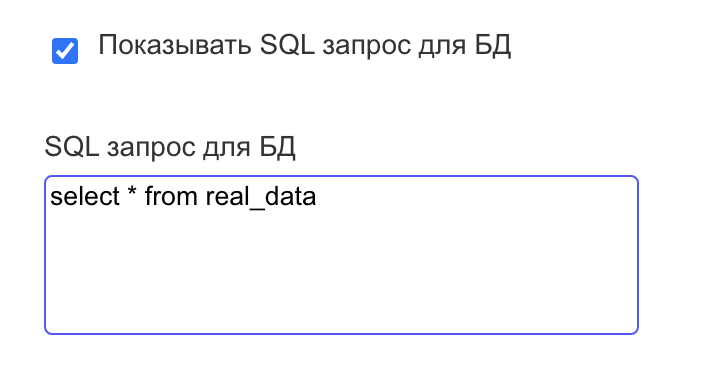
Начальный интерфейс



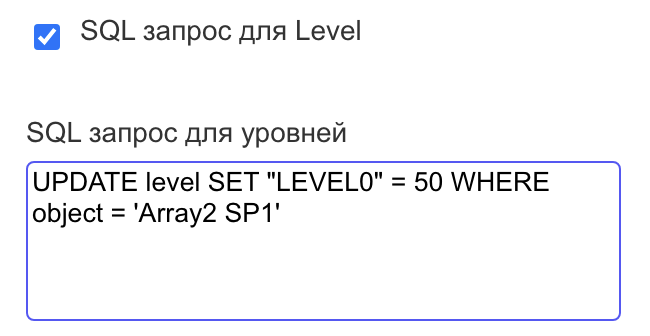
Выбор параметров



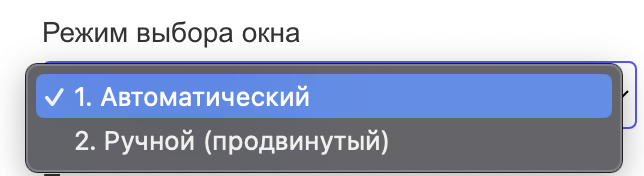
Чекбокс показать SQL запрос для БД и поле ввода запроса

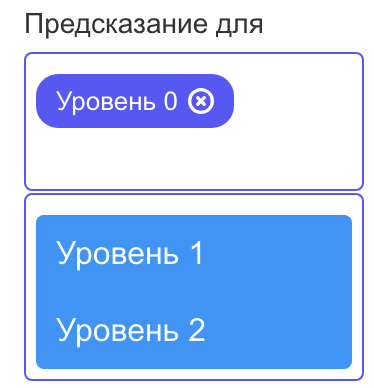


Чекбокс SQL запрос для Level и поле ввода запроса

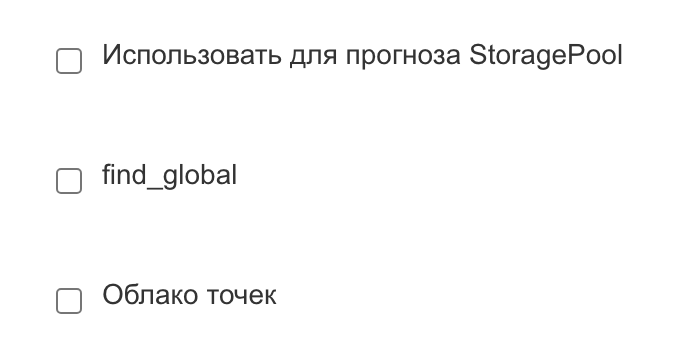


Режим выбора окна





Чекбоксы: использовать для прогноза StoragePool, find\_global и облако точек



Интерфейс после нажатия визуализировать

