

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
ТЕМА: БАЗА ДАННЫХ ГРИБОВ

Студенты гр. 6304

Некрасов Н.А.

Прозорова А.Д.

Иванкова В.М.

Преподаватель

Заславский М.М.

Санкт-Петербург

2019

ЗАДАНИЕ

Студенты

Некрасов Н.А.

Прозорова А.Д.

Иванкова В.М.

Группа 6304

Тема проекта: Разработка сайта для хранения базы данных грибов

Исходные данные:

Необходимо реализовать сайт для работы с информацией о грибах для СУБД MongoDB.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Качественные требования к решению»

«Сценарии использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложение»

«Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания:

Дата сдачи реферата:

Дата защиты реферата:

Студенты гр.6304

Некрасов Н.А.

Прозорова А.Д.

Иванкова В.М.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В рамках данного курса предполагалось разработать проект в команде на одну из поставленных тем. Была выбрана тема создания сайта для хранения базы данных грибов для СУБД MongoDB, так как был вызван интерес в связи с возможностью нахождения различных видов грибов на интересующей нас территории. Найти исходный код и всю дополнительную информацию можно по ссылке: <https://github.com/moevm/nosql1h19-mushrooms>

SUMMARY

As part of this course, it was planned to develop a project in a team on one of the topics. The topic of creating a site for storing a database of mushrooms for the MongoDB DBMS was chosen, as interest was aroused in connection with the possibility of finding different types of mushrooms in the territory of interest to us. Source code and all the additional information can be found at: <https://github.com/moevm/nosql1h19-mushrooms>

Содержание.

Введение.	5
Качественные требования к решению.	5
Сценарии использования.	6
Макет UI.	6
Описание сценариев использования.	10
Сценарий использования - «Поиск грибов по параметрам».	10
Сценарий использования - «Предложение гриба для занесения в основную коллекцию».....	10
Сценарий использования - «Рассмотрение предложений администратором».....	11
Сценарий использования - «Обновление/удаление записи из основной коллекции».	11
Сценарий использования - «Добавление записи в основную коллекцию».....	12
Модель данных.	12
NoSQL модель данных (MongoDB).	12
Коллекции.	12
Структура.....	12
Вычисление примерного объема данных.	13
Размер коллекций.....	14
Запросы.....	14
SQL модель данных.	14
Схема.....	14
Вычисление примерного объема данных.	15
Количество запросов.....	15
Сравнение SQL и NoSQL.	15
Разработанное приложение.	16
Краткое описание.	16
Схема экранов сайта.	17
Использованные технологии.....	18
Ссылки на Приложение.	18
Выводы.	18
Результаты.....	18
Недостатки и пути для улучшения полученного решения.	18
Приложение.	19
Документация по сборке и развертыванию приложения.	19
Снимки экранов приложения.	19
Список использованных источников.	22

Введение.

Цель работы – создать выгодное решение для хранения и анализа информации о грибах.

Было решено разработать веб-сайт, который позволит хранить всю информацию о грибах, позволяя при этом удобно с ней взаимодействовать.

Качественные требования к решению.

Требуется разработать сайт с использованием СУБД MongoDB.

Сценарии использования.

Макет UI.

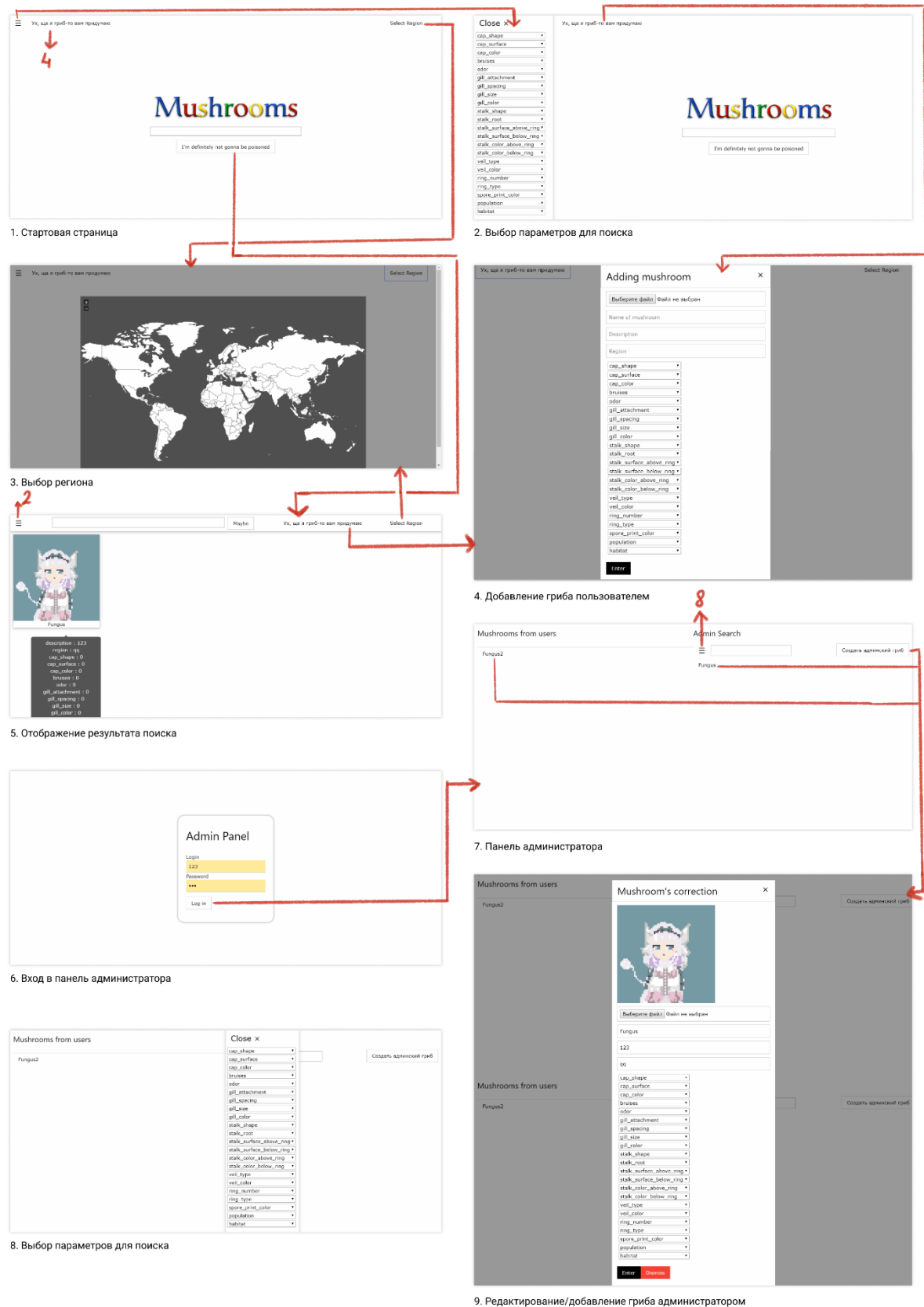


Рисунок 1 – Общий вид.

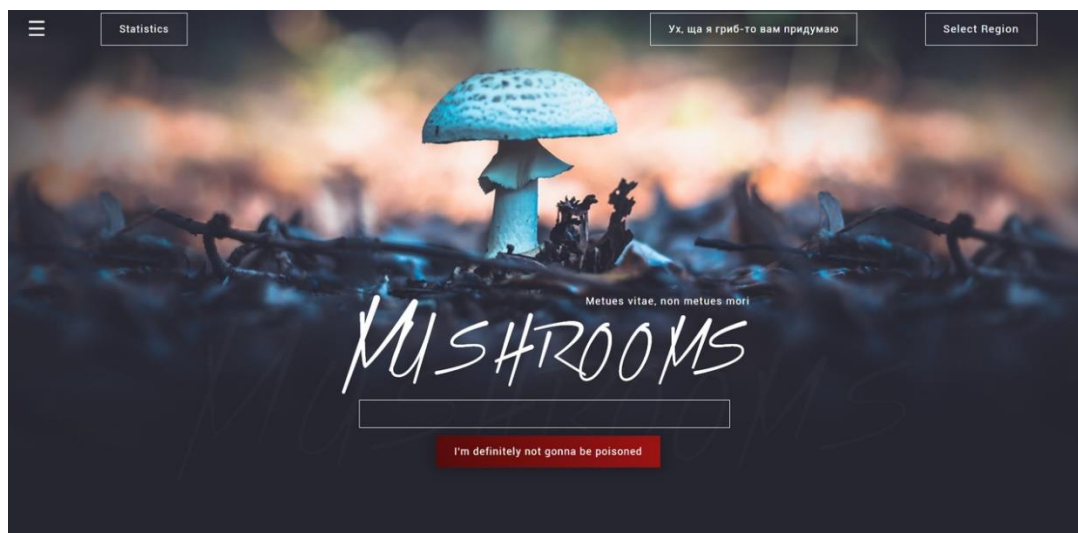


Рисунок 2 – Стартовая страница

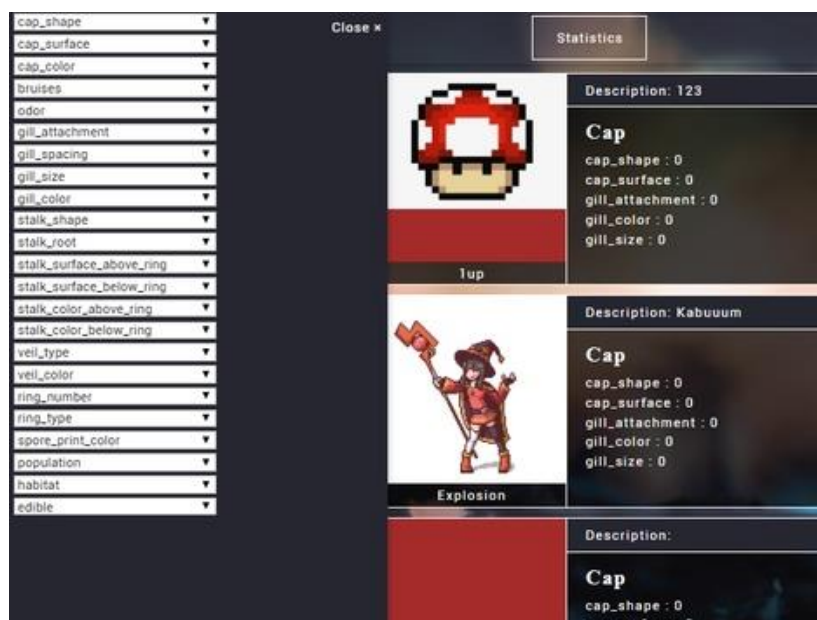


Рисунок 3 – Выбор параметров для поиска

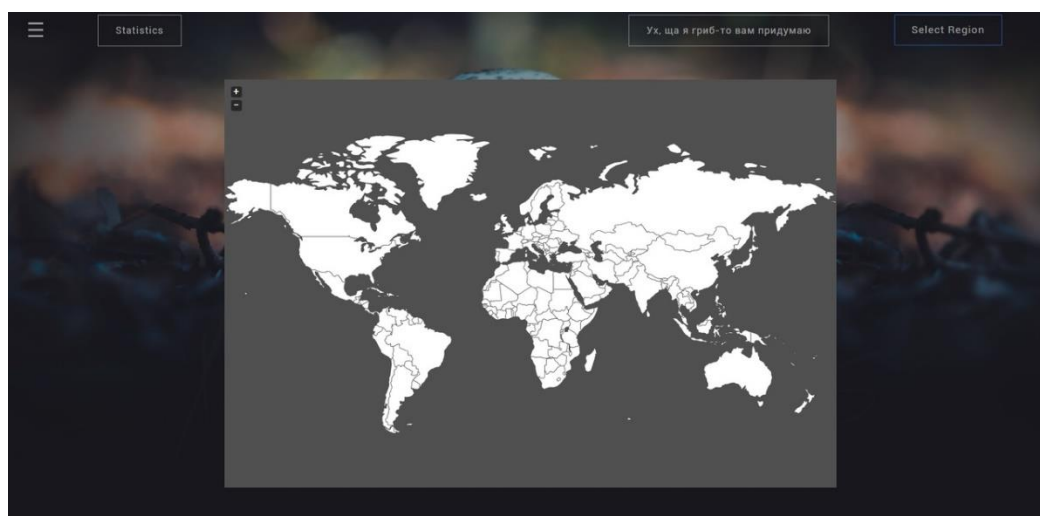


Рисунок 4 – Выбор региона



Рисунок 5 – Добавление гриба пользователем

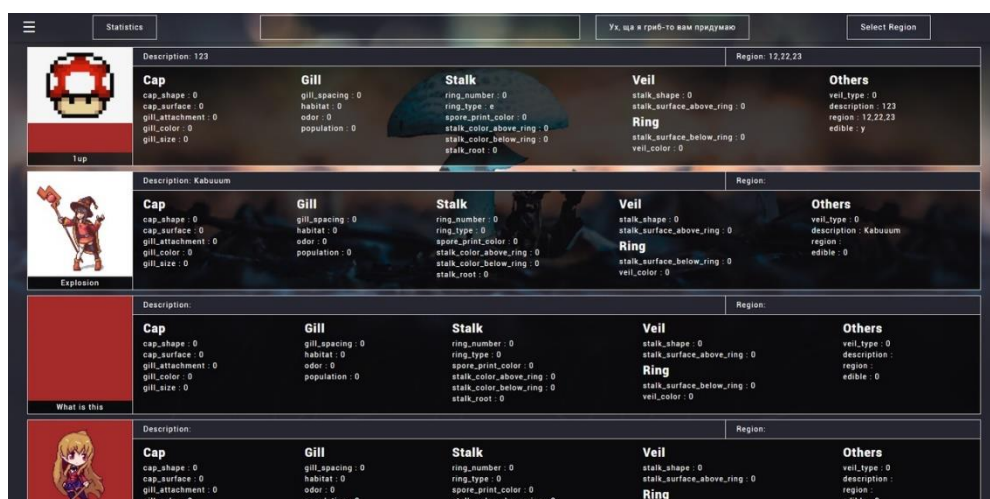


Рисунок 6 – Отображения результата поиска

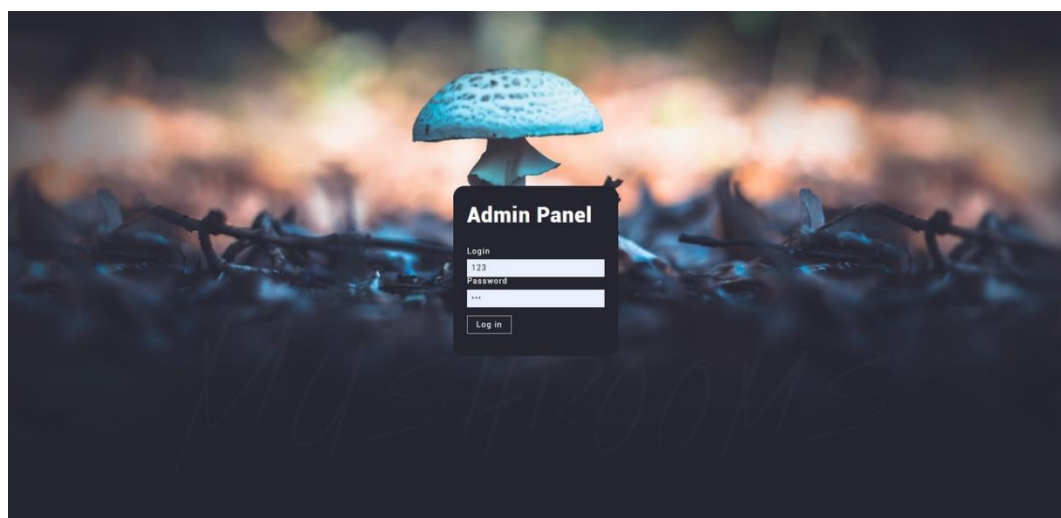


Рисунок 7 – Вход в панель администратора

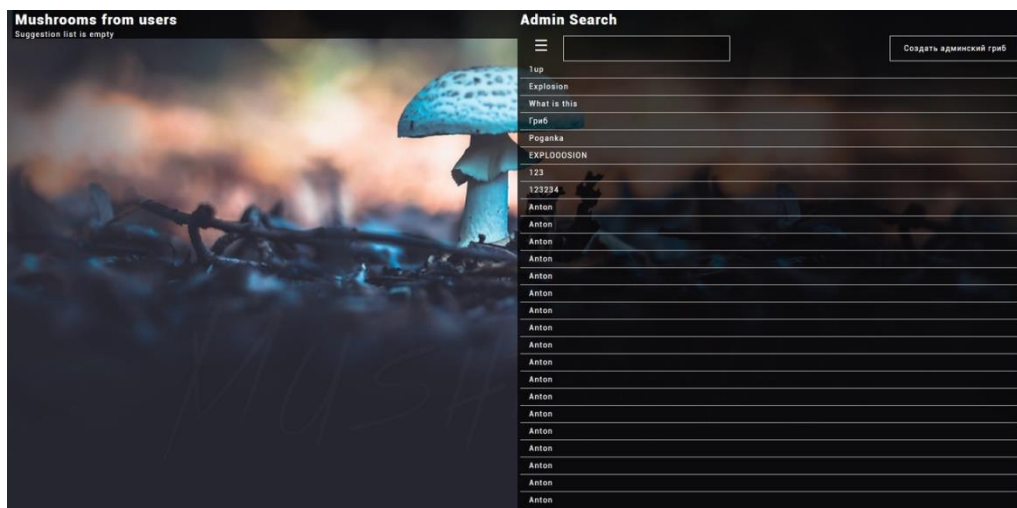


Рисунок 8 – Панель администратора

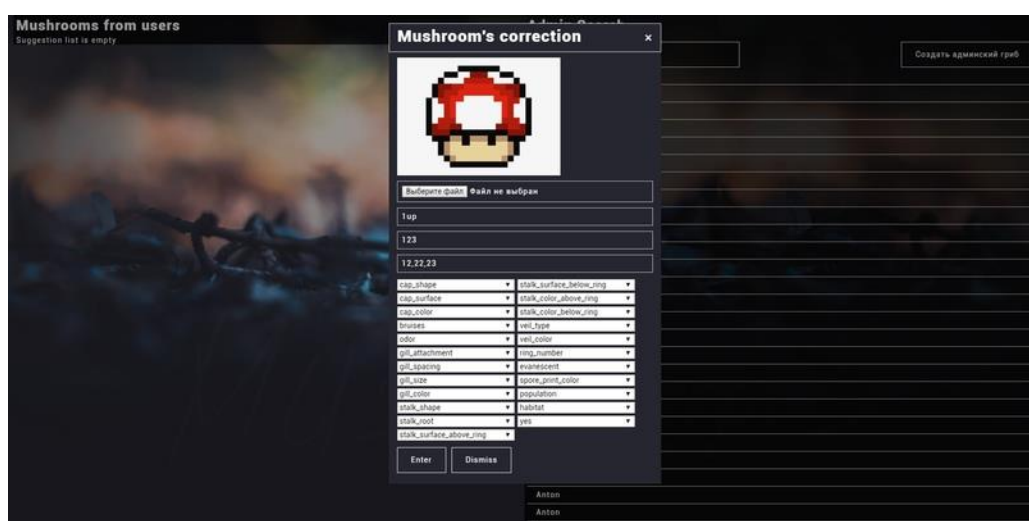


Рисунок 9 – Редактирование/добавление гриба администратором

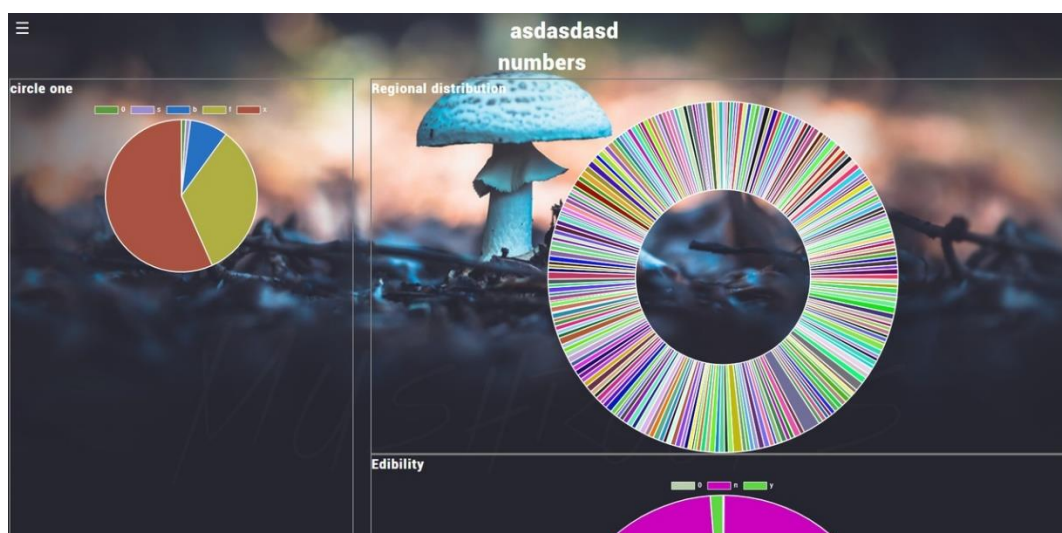


Рисунок 10 – Статистика

Описание сценариев использования.

Сценарий использования - «Поиск грибов по параметрам».

Действующее лицо: Пользователь.

Основной сценарий:

1. Пользователь заходит на сайт.
2. Вводит необходимые параметры, в качестве которых могут выступать:
 - 1) Регион произрастания
 - 2) Название
 - 3) Съедобность
 - 4) Специфические для гриба признаки, к примеру запах, цвет шляпки и т.д.
3. По нажатии на кнопку переносится на страницу с результатами, где может в реальном времени получать данные из БД.
4. При нажатии на карточку гриба появляется поп-ап с его характеристиками и описанием.
5. Пользователь читает предоставленную информацию.

Альтернативный сценарий:

- Предложение гриба для занесения в основную коллекцию.

Сценарий использования - «Предложение гриба для занесения в основную коллекцию».

Действующее лицо: Пользователь.

Основной сценарий:

1. Пользователь заходит на сайт.
2. Нажимает на кнопку «Ух, ща я гриб-то вам придумаю».
3. Заполняет открывшееся модальное окно.
4. По нажатии на кнопку отправляет на рассмотрение.

Альтернативный сценарий:

- Поиск грибов по параметрам.

Сценарий использования - «Рассмотрение предложений администратором».

Действующее лицо: Администратор.

Основной сценарий:

1. Администратор заходит на страницу /adminAuth.
2. Логинится под данным ему логином/паролем.
3. Переходит на страницу /adminpanel.
4. При нажатии на левый столбец, открывается модальное окно с параметрами, заполненными пользователем, после редактирования (при необходимости), администратор может нажать на чёрную кнопку для принятия предложения и отправления гриба в основную коллекцию, либо отклонить, нажатием на красную кнопку.

Альтернативный сценарий:

- Обновление/удаление записи из основной коллекции.
- Добавление записи в основную коллекцию.

Сценарий использования - «Обновление/удаление записи из основной коллекции».

Действующее лицо: Администратор.

Основной сценарий:

1. Администратор заходит на страницу /adminAuth.
2. Логинится под данным ему логином/паролем.
3. Переходит на страницу /adminpanel.
4. Нажатием на правый столбец открывается модальное окно с данными, после редактирования (при необходимости), администратор может нажать на черную кнопку для принятия гриба, или нажать на красную кнопку для удаления гриба.

Альтернативный сценарий:

- Добавление записи в основную коллекцию.
- Рассмотрение предложений администратором.

Сценарий использования - «Добавление записи в основную коллекцию».

Действующее лицо: Администратор.

Основной сценарий:

1. Администратор заходит на страницу /adminAuth.
2. Логинится под данным ему логином/паролем.
3. Переходит на страницу /adminpanel.
4. Нажатием на кнопку «Сделать админский гриб», открывает модальное окно и заполняет данные, нажимает на черную кнопку и отправляет запись в основную коллекцию.

Альтернативный сценарий:

- Рассмотрение предложений администратором.
- Обновление/удаление записи из основной коллекции.

Модель данных.

NoSQL модель данных (MongoDB).

Коллекции.

Данные хранятся с использованием MongoDB в двух коллекциях:

- Mushrooms – коллекция, в которой хранится основной массив записей о грибах
- Suggestions – коллекция, в которую пользователи могут отправить записи на добавление записи в основную коллекцию

Структура.

Документы обладают идентичной структурой:

- _id: ObjectId
- bruises: String
- cap_color: String
- cap_shape: String
- cap_surface: String
- gill_attachment: String

- gill_color: String
- gill_size: String
- gill_spacing: String
- habitat: String
- name: String
- odor: String
- population: String
- ring_number: String
- ring_type: String
- spore_print_color: String
- stalk_color_above_ring: String
- stalk_color_below_ring: String
- stalk_root: String
- stalk_shape: String
- stalk_surface_above_ring: String
- stalk_surface_below_ring: String
- veil_color: String
- veil_type: String
- description: String
- region: String
- img: String

Вычисление примерного объема данных.

$$V(doc) = \text{summ}(V(field))$$

Fields:

_id - 12b;

22 поля - 1 символ - 44b;

name ~ 15 символов - 30b;

description ~ 30 символов - 60b;

region ~ 3 значения в среднем - 6 символов - 12b;

img ~ 8000+ символов - 16000b;

$$V(doc) = \text{summ}(V(field)) \sim 16158b \sim 15.8Kb$$

Размер коллекций.

$$V(collection) = V(doc) * N, \text{ where}$$

N — size of a single doc

Исходя из разных источников (<https://en.wikipedia.org/wiki/Mushroom>,
http://www.ikonet.com/en/visualdictionary/static/us/edible_mushrooms)

фунгусов насчитывается ~ 50'000

$$V(collection) \sim V(doc) * 50000 \sim 790'000Kb \sim 771.48Mb$$

Запросы.

N - количество документов в коллекции.

- Добавление гриба в основную коллекцию/предложения

Один запрос

```
Mushrooms.save({name: "Fungus", ...});  
Suggestions.save({name: "Лисичка", ...});
```

- Удаление гриба по `_id` из основной коллекции/предложений

На `_id` стоит уникальный индекс, значит количество запросов $\log_2(N)$

```
Mushrooms.deleteOne({"_id": data._id}, ...)  
Suggestions.deleteOne({"_id": data._id}, ...)
```

- Перенос между коллекциями

a. Сохранение в основную - один запрос

b. Удаление по `_id` из предложений - $\log_2(N)$ запросов

- Поиск по имени/региону

На `name/region` стоит индекс, количество запросов - $\log_2(N)$

- Поиск по параметрам

На данный момент - не оптимизировано, поэтому $O(N)$

SQL модель данных.

Схема.

Для хранения потребовалось бы 4 таблицы в четвёртой нормальной форме.

Для каждой из коллекций потребовалось бы:

- Таблица для хранения всех параметров за исключением региона

- Таблица `_id`, `region`

При этом пары между собой связаны бы не были.

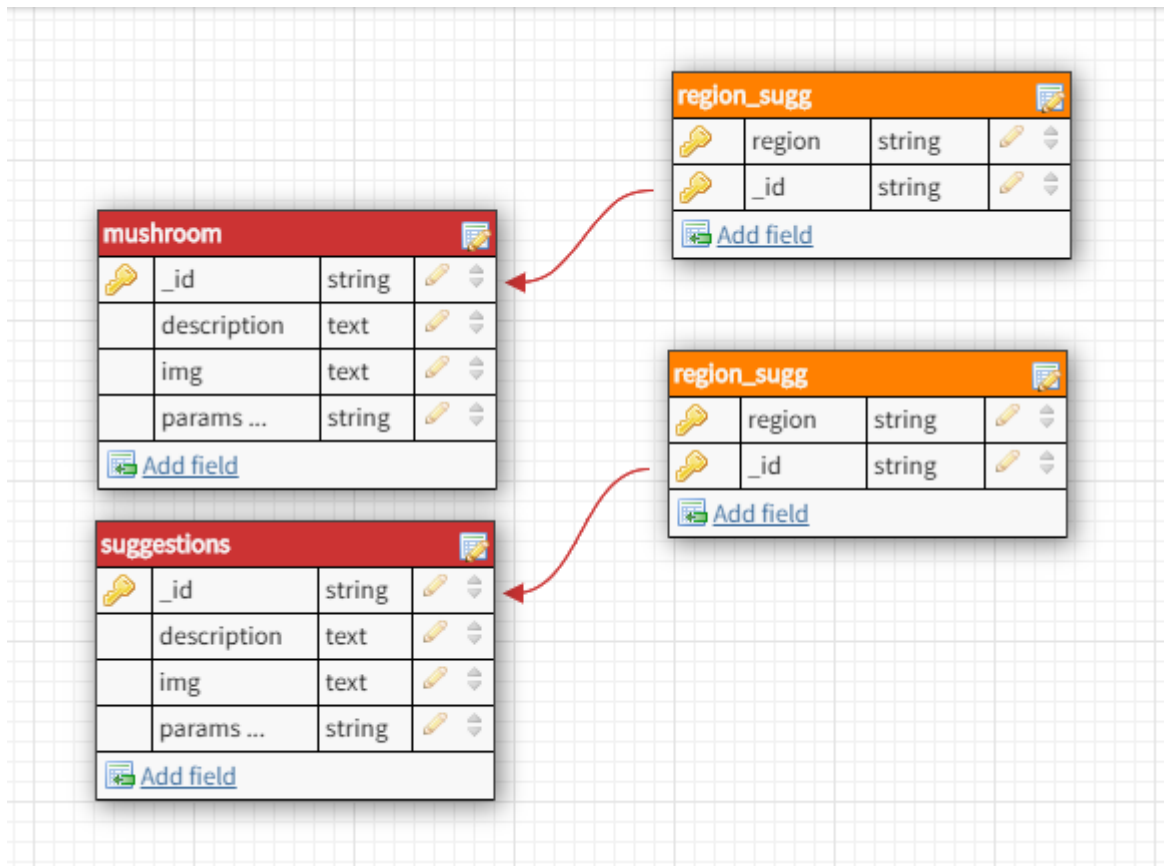


Рисунок 11 – Модель данных SQL

Вычисление примерного объема данных.

N - количество грибов таблице

Аналогичен NoSQL + $(24b * N) * 2$ - дополнительные две таблицы

Количество запросов.

Для каждого из юз-кейсов увеличивается на $\log_2(n)$, где n - количество `_id` в таблице регионов

Для запроса на перенос значения $\log_2(n) + 1$

Сравнение SQL и NoSQL.

Использование NoSQL БД выгоднее по ряду причин:

- Поскольку NoSQL позволяет хранить в качестве поля массив, не требуются дополнительные таблицы для хранения регионов

- Ниже количество запросов, ввиду того, что в SQL версии требуется дополнительных $\log_2(n)$ запросов на получение региона для нужной записи из связанной таблицы
- Горизонтальная масштабируемость

Разработанное приложение.

Краткое описание.

Веб-сайт, хранящий в себе грибы, их съедобность, специфические для гриба признаки, регионы произрастания и связи между ними, с возможностями анализа имеющейся информации о них.

Схема экранов сайта.

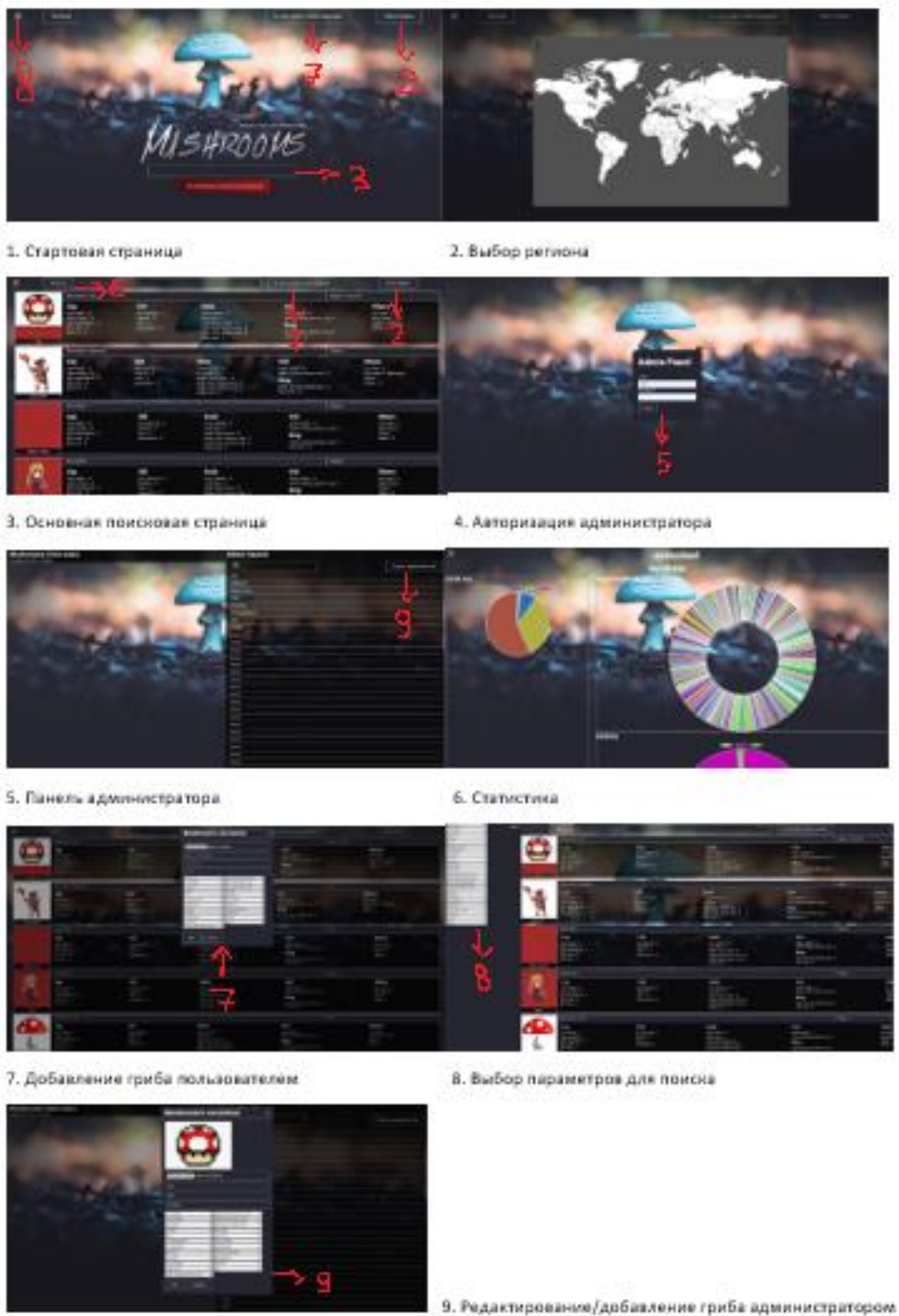


Рисунок 12 – Схема экранов сайта

Использованные технологии.

- Node.js
- MongoDB
- JQuery
- passport.js
- w3-css
- bootstrap
- express.js
- body-parser
- cookie-express
- mongoose
- cookie-session
- pug

Ссылки на Приложение.

Ссылка на github: <https://github.com/moevm/nosqlh19-mushrooms>

Выводы.

Результаты.

В ходе работы было разработано приложение для анализа информации о базе данных грибов.

Недостатки и пути для улучшения полученного решения.

На данный момент не оптимизировано получение данных из БД, ввиду того что, результаты предыдущих запросов не учитываются в формировании следующих запросов.

Приложение.

Документация по сборке и разворачиванию приложения.

1. Скачать проект из репозитория
2. Запустить `www/bin`
3. Установить администраторский пароль в `login/index`

Снимки экранов приложения.

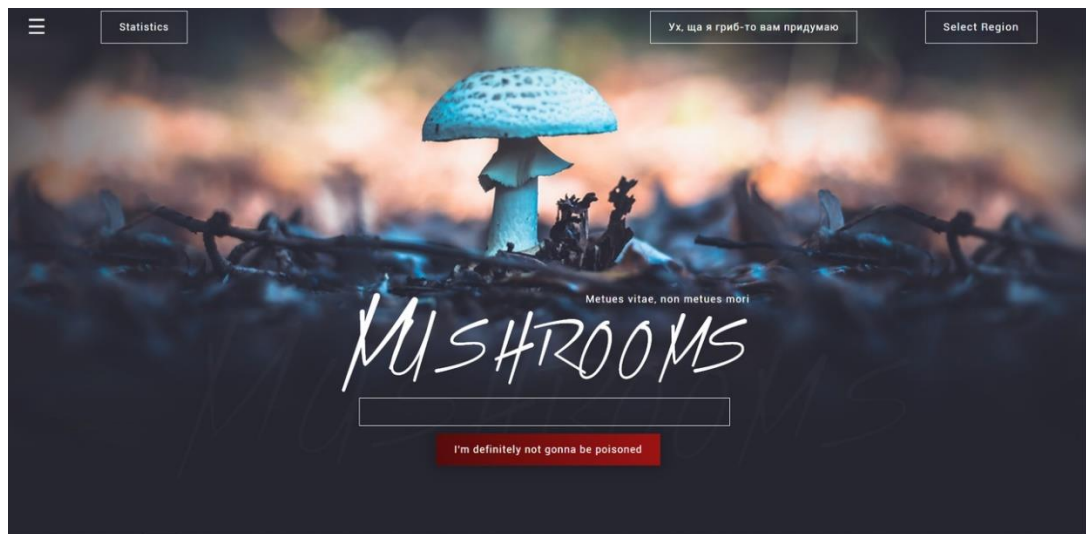


Рисунок 13 – Стартовая страница

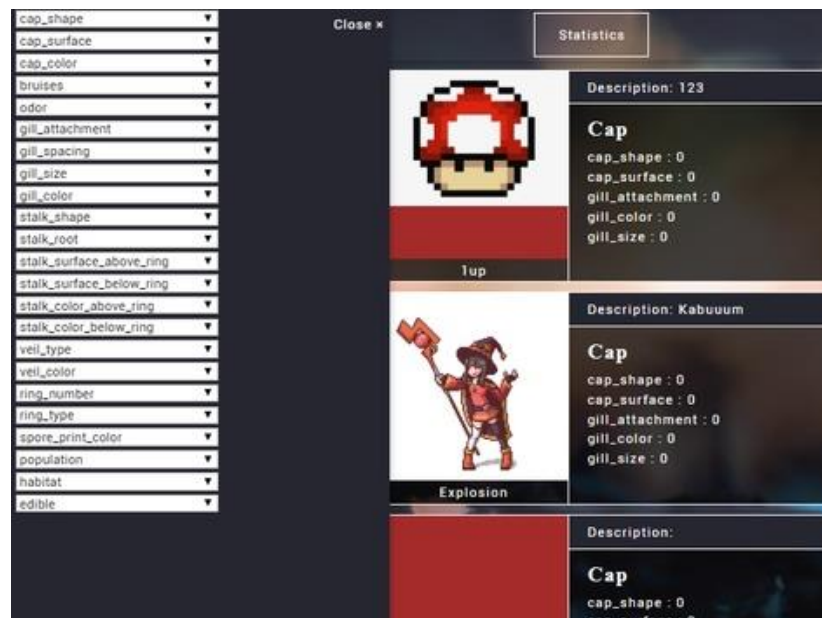


Рисунок 14 – Выбор параметров для поиска

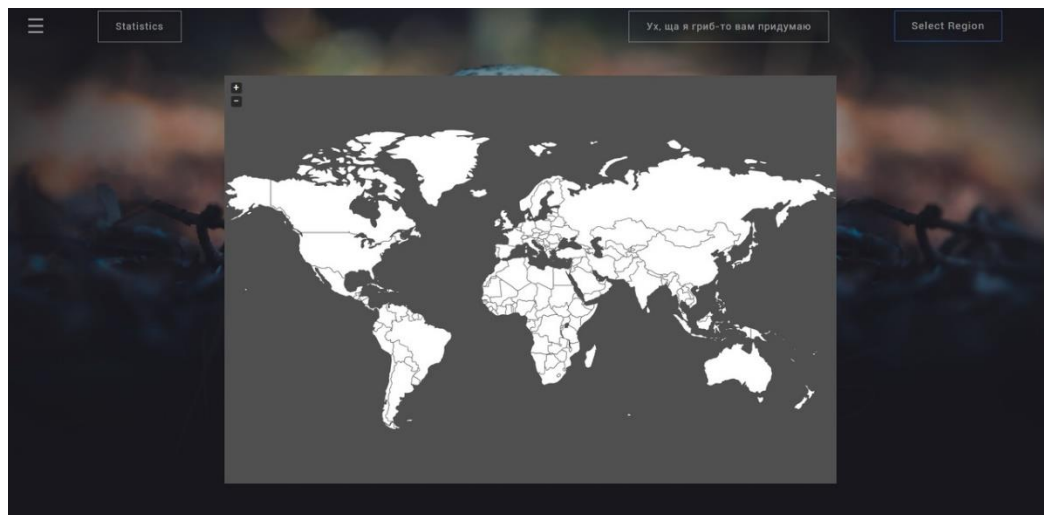


Рисунок 15 – Выбор региона

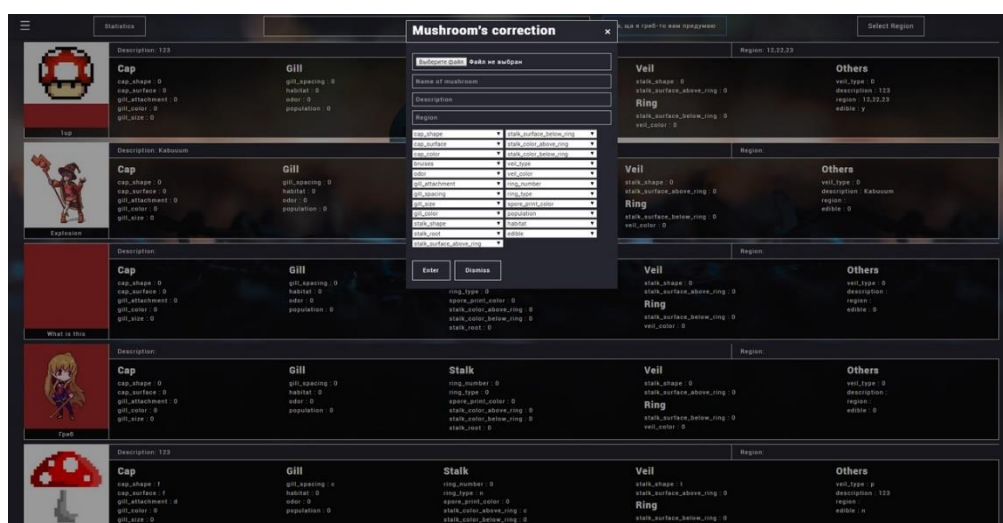


Рисунок 16 – Добавление гриба пользователем

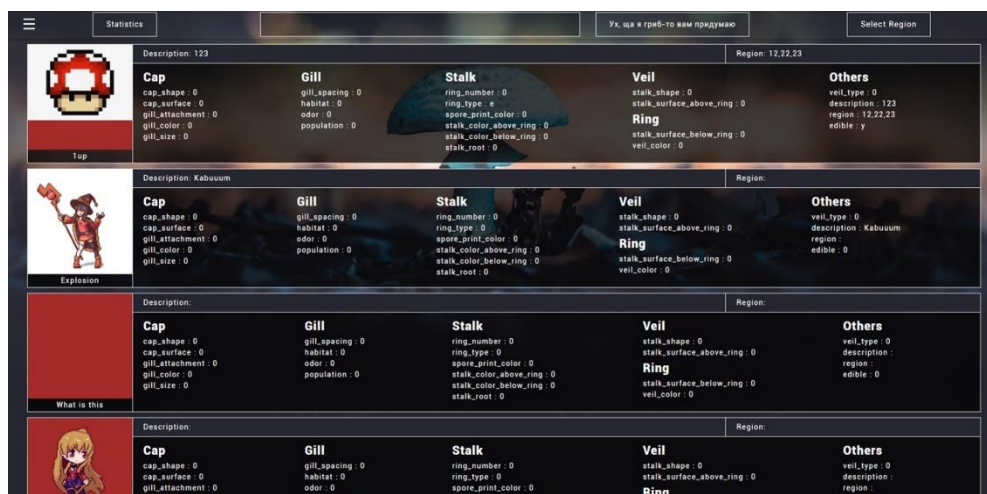


Рисунок 17 – Отображение результата поиска

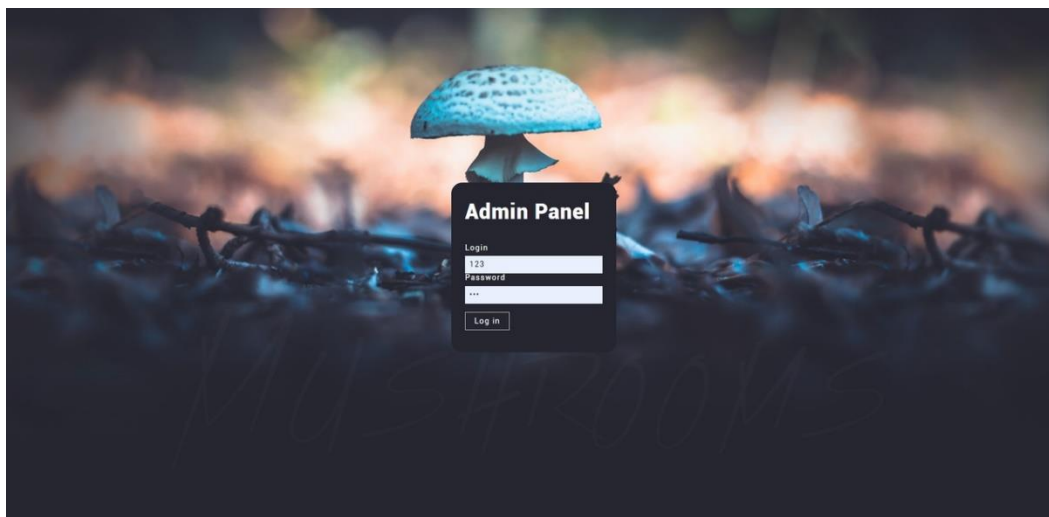


Рисунок 18 – Вход в панель администратора

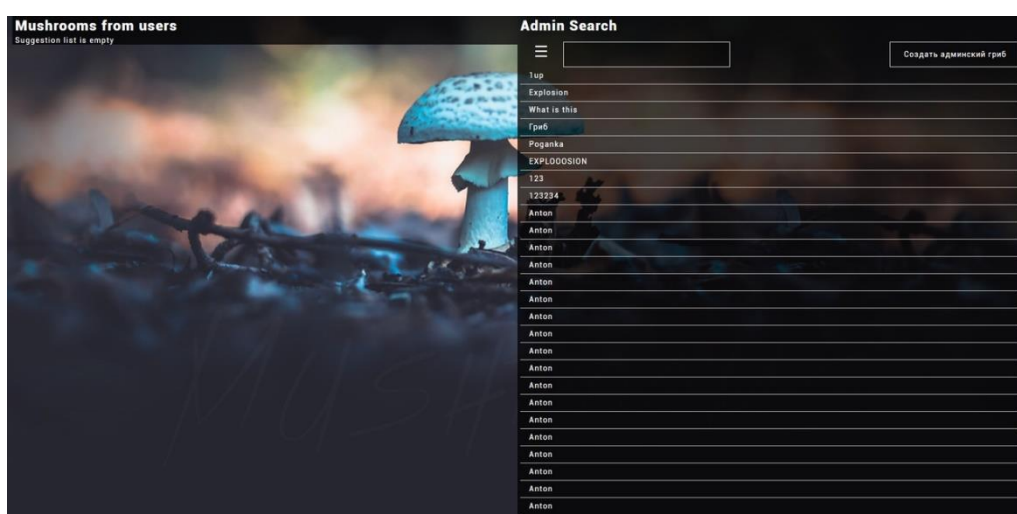


Рисунок 19 – Панель администратора

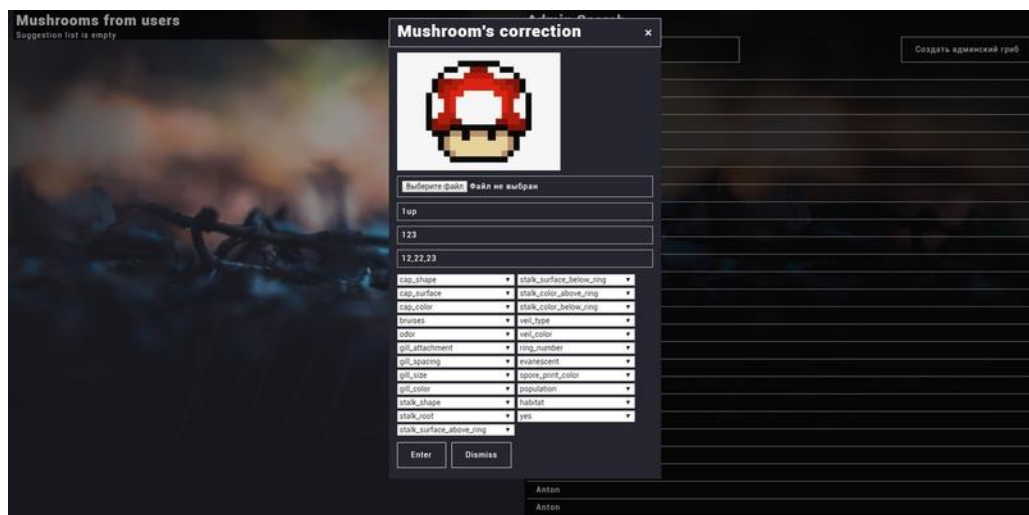


Рисунок 20 – Редактирование/добавление гриба администратором

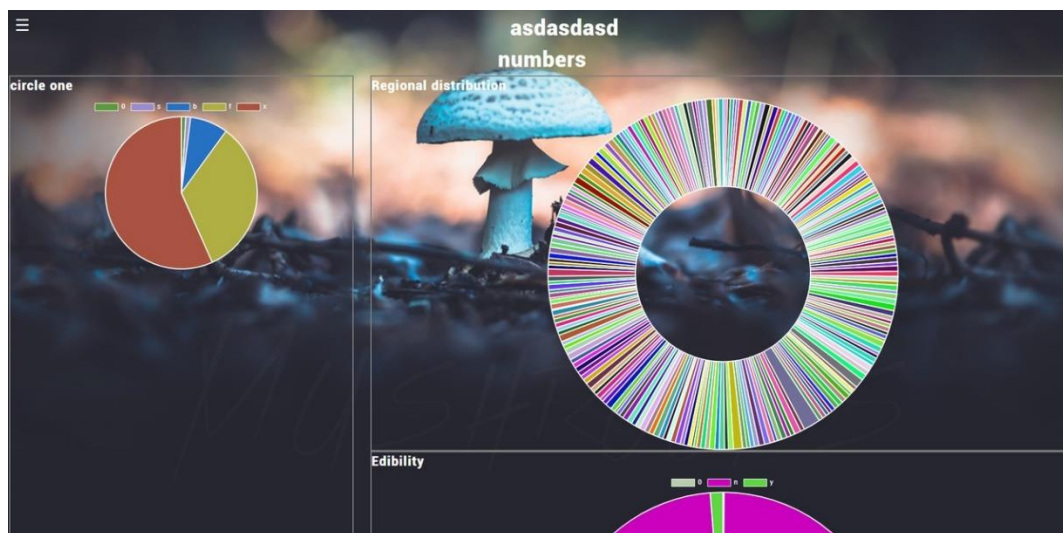


Рисунок 21 – Статистика

Список использованных источников.

1. Документация MongoDB: <https://docs.mongodb.com/>
2. Репозиторий проекта: <https://github.com/moevm/nosql1h19-mushrooms>