Разработка веб-сайта для мониторинга ситуации с больными коронавирусной инфекций COVID-19

Как сообщает ВОЗ, 31 декабря 2019 г. было получено сообщение о группе случаев заболевания «вирусной пневмонией» в городе Ухане, Китайская Народная Республика. Теперь известно, что новое заболевание, называемое COVID-19, было вызвано новым коронавирусом, оказавшимся очень агрессивным и жизнеспособным. Вспышка переросла в мировую пандемию, которая до сих пор с трудом идет на спад, при этом в ученом сообществе существуют разногласия и единого мнения о том, как управлять ситуацией – нет.

В этих условиях очень важно обеспечить людей информацией о ситуации с заболеванием в мире, позволить анализировать ее и делать выводы, чтобы каждый мог принять меры безопасности для себя и своих близких.

Итак, **цель моей работы** - разработать базу данных, содержащую информацию о больных коронавирусной инфекцией и пользователях веб-сайта, и веб- приложение для ее анализа.

Чтобы достигнуть поставленной цели, требуется решить следующие **задачи**.

1. Проанализировать предметную область, сформулировать ограничения предметной области.
2. Провести анализ существующих СУБД.
3. Спроектировать базу данных, содержащую информацию о больных коронавирусной инфекцией и пользователях веб-сайта.
4. Реализовать веб-приложение для анализа спроектированной базы данных.

**Требования к программе**

Определим возможности всей системы.

1. Регистрация.
2. Аутентификация/Авторизация.

Веб- сайт должен предоставлять возможность авторизации как с правами обычного пользователя, так и с правами, предоставляющими больший доступ к базе данных.

1. Просмотр таблиц базы данных, относящихся к зарегистрированным случаям заражения COVID-19.
2. Выбор запроса к базе данных из предложенного списка запросов для получения результата.
3. Создание личного кабинета зарегистрированного пользователя.
4. Фиксация состояния здоровья в личном кабинете (Здоров/Болен COVID- 19).
5. Просмотр статистики заболевших COVID-19 пользователей, проживающих на той же улице, что и авторизованный пользователь, в его личном кабинете.
6. Дополнительные возможности специалиста:

8.1. Просмотр дополнительных сведений о таблицах базы данных: название таблицы, схема таблицы.

8.2. Просмотр развязочной таблицы.

8.3. Просмотр диаграммы «сущность-связь».

Эти сведения потребуются специалисту при составлении запросов и внесении изменений в БД.

8.4. Создание запроса к базе данных.  
 8.5. Внесение изменений в базу данных.

Предполагается, что специалист владеет информацией и языком запросов SQL, чтобы составить релевантный запрос или внести изменение, повышающее актуальность данных.

**Формализация данных**

База данных должна содержать информацию о следующих объектах.

1. Пользователи сайта:

1.1. Данные для аутентификации.

2.2. Персональные данные для личного кабинета.

1. Улицы Москвы.
2. Запросы к базе данных, имеющиеся в системе и созданные специалистами.
3. Пациенты с зарегистрированными случаями заражения COVID-19.
4. Местоположение пациентов.
5. Симптомы, наблюдающиеся при COVID-19.
6. Статус состояния здоровья пациентов.

**Сведения о данных**

Более подробно об этих объектах скажет следующая таблица. Здесь представлены данные, которые хранит каждый объект.

|  |  |
| --- | --- |
| **Категория** | **Сведения** |
| Пользователь | Email-адрес, пароль |
| Личный кабинет | Владелец кабинета, имя, фамилия, дата рождения, пол, адрес проживания (название улицы), состояние здоровья |
| Роль | Название, описание |
| Улица | Название, численность населения, количество больных COVID-19 на этой улице |
| Запросы | Название, описание, SQL-запрос, названия выходных столбцов |
| Пациенты | Дата регистрации в базе данных, пол, возраст, дата начала симптомов, дата посещения больницы, дата начала наблюдения, дата окончания наблюдения, посещал ли пациент Ухань, является ли пациент жителем Уханя |
| Местоположение | Провинция/штат, страна, широта, долгота |
| Симптомы | Название |
| Статус состояния здоровья | Название |

**Диаграмма прецедентов**

Информацию о видах пользователей и доступных им возможностях можно представить на диаграмме прецедентов.

|  |  |
| --- | --- |
| **Вид пользователя** | **Возможности** |
| Неавторизованный | Регистрация, авторизация  Просмотр таблиц базы данных (всех, кроме развязочной) |
| Авторизованный (обычный пользователь) | Создание личного кабинета, фиксация статуса состояния здоровья (Здоров/Болен COVID-19)  Просмотр статистики заболевших на своей улице проживания  Просмотр запросов (выбор любого из списка для получения результата)  Просмотр таблиц базы данных (всех, кроме развязочной) |
| Авторизованный  (специалист) | Создание личного кабинета, фиксация статуса состояния здоровья (Здоров/Болен COVID-19)  Просмотр статистики заболевших на своей улице проживания  Просмотр запросов (выбор любого из списка для получения результата)  Создание запросов  Просмотр таблиц базы данных и ER-диаграммы  Внесение изменений в базу данных |

**Выбор СУБД**

Мной были рассмотрены 3 типа СУБД по модели данных.

1. Дореляционные, которые, в свою очередь, подразделяются на:

1.1. Системы, основанные на инвертированных списках

1.2. Иерархические СУБД

1.3. Сетевые СУБД

Основные недостатки СУБД, основанных на дореляционных моделях данных, это высокая сложность схемы базы данных, построенной на основе этих моделей, негибкость, избыточность и большой расход ресурсов памяти.

2. Реляционные

Реляционная модель представляет собой совокупность данных, состоящую из набора двумерных таблиц

Реляционная модель является удобной и наиболее привычной формой представления данных. При табличной организации данных, отсутствует иерархия элементов. Строки и столбцы могут быть просмотрены в любом порядке, поэтому высока гибкость выбора любого подмножества элементов в строках и столбцах.

3. Постреляционные

Также была рассмотрена постреляционная модель. Она использует трехмерные структуры, позволяя хранить в полях таблицы другие таблицы, расширяя таким образом возможности по описанию сложных объектов реального мира. Но в таких СУБД сложно обеспечить целостность данных.

Таким образом, так как задача предполагает использование разнообразных запросов, включающее в себя выборки элементов строк и столбцов различной сложности, приоритетным свойством модели данных является гибкость, простота использования и независимость от приложения, поэтому в качестве модели организации данных была выбрана реляционная модель.

**Проектирование базы данных**

В соответствии с формализацией данных, были выделены следующие таблицы и схемы.