

Разработка блога на Express и MongoDB с использованием Handlebars

Цель задания: Создать веб-приложение блога, где пользователи могут читать статьи, а администраторы — добавлять, редактировать и удалять их. В процессе выполнения задания вы закрепите знания по работе с Express, MongoDB, шаблонизатором Handlebars, а также научитесь использовать переменные окружения и работать с middleware.

Основные требования:

1. Настройка проекта:

- Инициализировать новый проект с помощью NPM, PNPM или Yarn.
- Установить необходимые зависимости: `express`, `mongoose`, `express-handlebars`, `dotenv` и любые другие, которые посчитаете необходимыми.

2. Работа с базой данных:

- Подключиться к MongoDB с использованием Mongoose. Использовать переменные окружения для хранения строки подключения.
- Создать схему и модель для статей с полями: заголовок, содержание, дата публикации.

3. Разработка маршрутов:

- Реализовать CRUD операции для статей:
 - **Create:** Форма для добавления новой статьи.
 - **Read:** Отображение списка всех статей и отдельной статьи.
 - **Update:** Форма для редактирования статьи.
 - **Delete:** Удаление статьи.
- Добавить страницу администратора, доступную только при наличии определённой переменной окружения (например, `isAdmin=true`).

4. Использование шаблонизатора Handlebars:

- Настроить Handlebars в качестве шаблонизатора для Express.
- Создать основные шаблоны и частичные шаблоны для повторно используемых компонентов, таких как хедер и футер.
- Использовать хелперы для форматирования дат.

5. Применение middleware:

- Реализовать middleware для логирования запросов.
- Создать middleware для проверки переменной окружения `isAdmin` перед доступом к странице администратора.

6. Безопасность и развертывание:

- Добавить в `.gitignore` файл `.env` и другие чувствительные данные.
- (Дополнительно) Развернуть приложение на любом облачном сервисе (например, Heroku).

7. Маршрутизация

- 1. Модификация модели статьи (если требуется):** Убедитесь, что ваша модель статьи содержит все необходимые поля, включая ID, который автоматически генерируется MongoDB.
- 2. Создание маршрута для отдельной статьи:** В файле `app.js` добавьте новый маршрут, который будет обрабатывать GET-запросы к отдельным статьям по их ID. Внутри обработчика маршрута используйте метод `findById` модели `Article` для поиска статьи по ID, полученному из параметров маршрута.
- 3. Отображение отдельной статьи:** Создайте новый шаблон Handlebars (например, `article.handlebars`), который будет отвечать за отображение отдельной статьи. В этом шаблоне вы можете использовать объект статьи, переданный из обработчика маршрута, для отображения заголовка, содержания, даты публикации и любой другой соответствующей информации о статье.
- 4. Добавление ссылок на отдельные статьи:** В шаблоне, который отображает список всех статей (например, в `home.handlebars` или `articles.handlebars`), добавьте ссылки на отдельные статьи, используя ID каждой статьи в качестве части URL. Это позволит пользователям переходить к просмотру полного текста статьи, кликнув по заголовку или кнопке "Читать далее".
- 5. Обработка отсутствующих статей:** В обработчике маршрута для отдельной статьи добавьте логику для обработки случаев, когда статья с указанным ID не найдена. Вы можете отобразить сообщение об ошибке или перенаправить пользователя на страницу с общим списком статей.



Больше не могу
работать, хочу
отдыхать