Project 2

Blog Post Manager

Description: A blogging platform where users can create, view, edit, and delete blog posts, now with distinct pages for creating posts, listing all posts, viewing post details, and editing posts. The application uses MySQL for data storage and React for the frontend with multiple routes.

Key Features:

- **Create Post Page**: Form to add a new blog post with title, content, and author.
- **Posts List Page**: Displays all blog posts with options to view details or delete.
- **Post Details Page**: Shows full details of a single post with options to edit or delete.
- **Edit Post Page**: Form to update an existing post (optional, but included for completeness).
- Backend: Store posts in a MySQL database.
- Frontend: React components with React Router for navigation.

Backend (Node.js):

- Use Express.js with mysql2.
- Create a posts table:
```
CREATE TABLE posts (
      id INT AUTO_INCREMENT PRIMARY KEY,
      title VARCHAR(255) NOT NULL,
      content TEXT NOT NULL,
      author VARCHAR(100) NOT NULL,
      created_at DATETIME DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```
- Implement APIs with validation.

API Endpoints:

1. **POST /posts** - Create a new post

   Request:
```
{
  "title": "My First Post",
  "content": "This is the content...",
  "author": "John Doe"
}
```

Response (201 Created):

```
{
  "id": 1,
  "title": "My First Post",
  "content": "This is the content...",
  "author": "John Doe",
  "created_at": "2025-04-30T12:00:00Z"

}
```

Error (400 Bad Request):

```
{ "error": "Title, content, and author are required" }
```

2. **GET /posts** - Retrieve all posts

Request: None

Response (200 OK):

```
[
  {
    "id": 1,
    "title": "My First Post",
    "content": "This is the content...",
    "author": "John Doe",
    "created_at": "2025-04-30T12:00:00Z"
  }
]
```

3. **GET /posts/:id - Retrieve a single post (new for Post Details page)**

Request: URL parameter id (e.g., /posts/1)

Response (200 OK):

```
{

  "id": 1,

  "title": "My First Post",

  "content": "This is the content...",

  "author": "John Doe",
  "created_at": "2025-04-30T12:00:00Z"
}
```

4. **Error (404 Not Found):**

   ```
   { "error": "Post not found" }
   ```

5. **PUT /posts/:id** - Update a post

   Request:

   ```
   {
     "title": "Updated Post",
     "content": "Updated content...",
     "author": "John Doe"
   }
   ```

   Response (200 OK):

   ```
   {
     "id": 1,
     "title": "Updated Post",
     "content": "Updated content...",
     "author": "John Doe",
     "created_at": "2025-04-30T12:00:00Z"
   }
   ```

   Error (404 Not Found):

   ```
   { "error": "Post not found" }
   ```

6. **DELETE /posts/:id** - Delete a post

   Request: None

   Response (204 No Content): Empty

   Error (404 Not Found):

   ```
   { "error": "Post not found" }
   ```

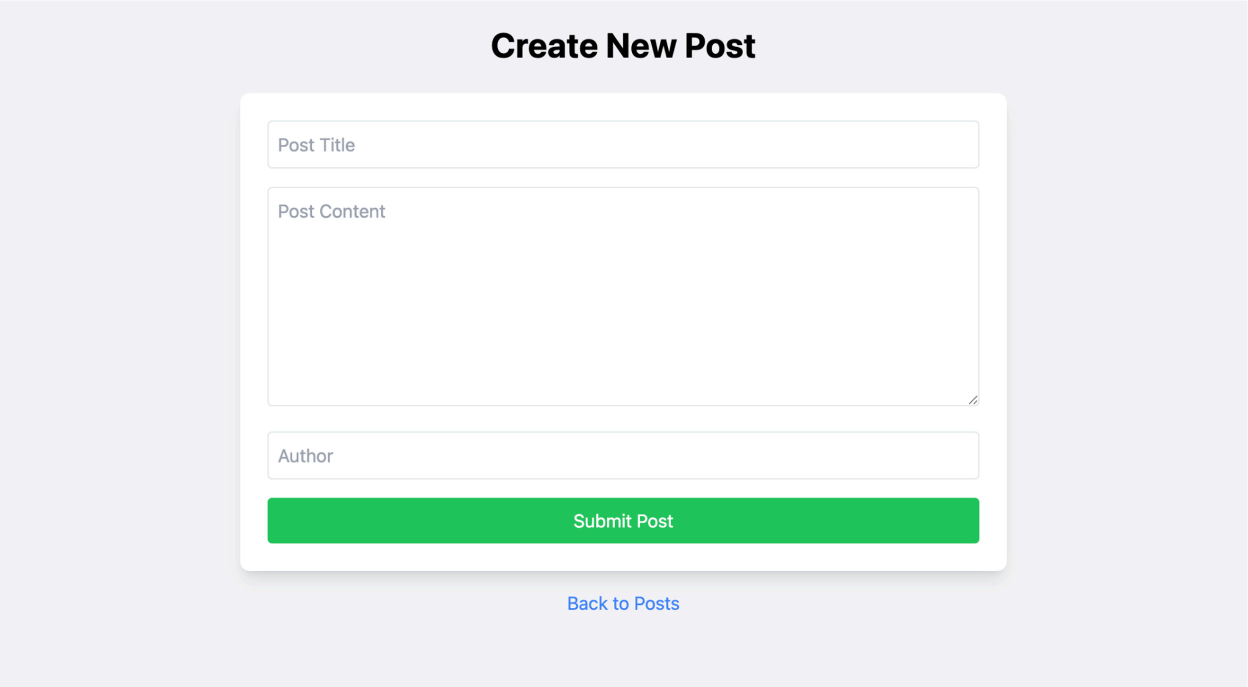
CRUD Queries:

- **Create:** `INSERT INTO posts (title, content, author) VALUES (?, ?, ?);`
- **Read:** `SELECT * FROM posts;`
- **Update:** `UPDATE posts SET title = ?, content = ?, author = ? WHERE id = ?;`
- **Delete:** `DELETE FROM posts WHERE id = ?;`

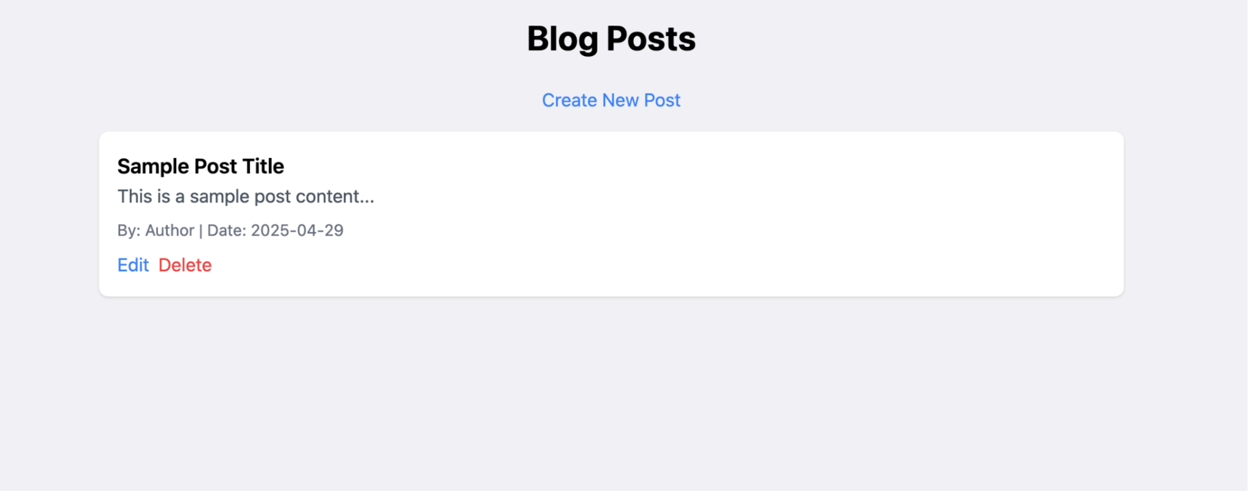
Frontend (React):

- Use React Router to manage routes:
    - **/create:** Create Post page
    - **/**: Posts List page
    - **/posts/:id:** Post Details page
    - **/posts/:id/edit:** Edit Post page
- Components:
    - **PostForm:** Reusable for creating and editing posts (handles form input for title, content, author).
    - **PostList**: Displays posts with links to details and delete buttons.
    - **PostDetails**: Shows full post content with edit/delete options.
- Use Axios for API calls and React hooks (useState, useEffect, useParams, useNavigate) for state and navigation.
- Implement basic form validation (e.g., ensure title and content are not empty).

Design:



Create New Post

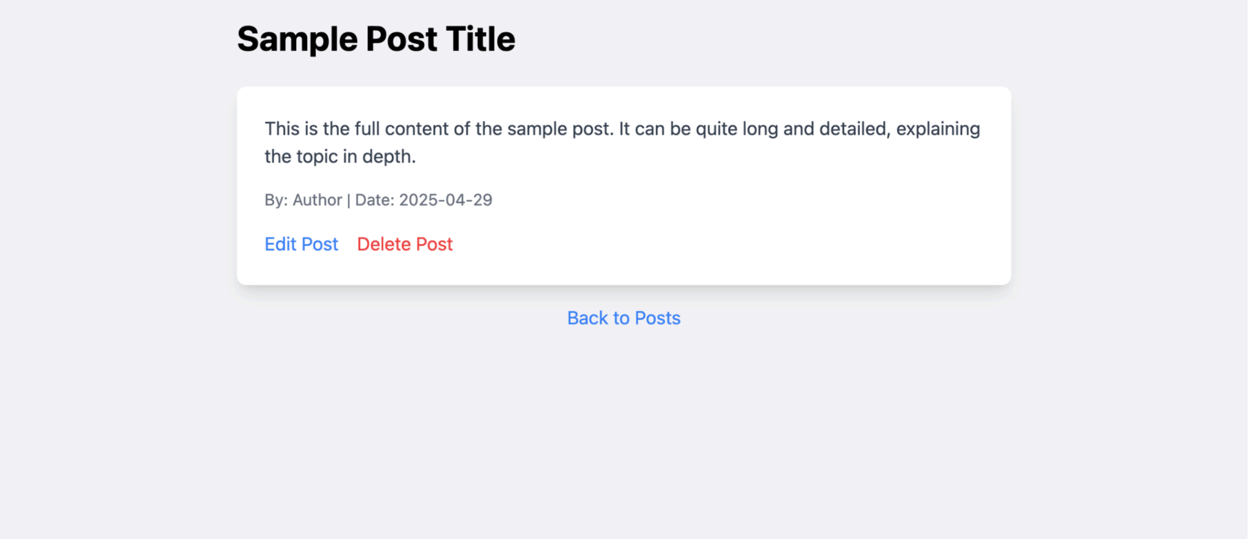


Blog Posts

Sample Post Title

This is a sample post content...

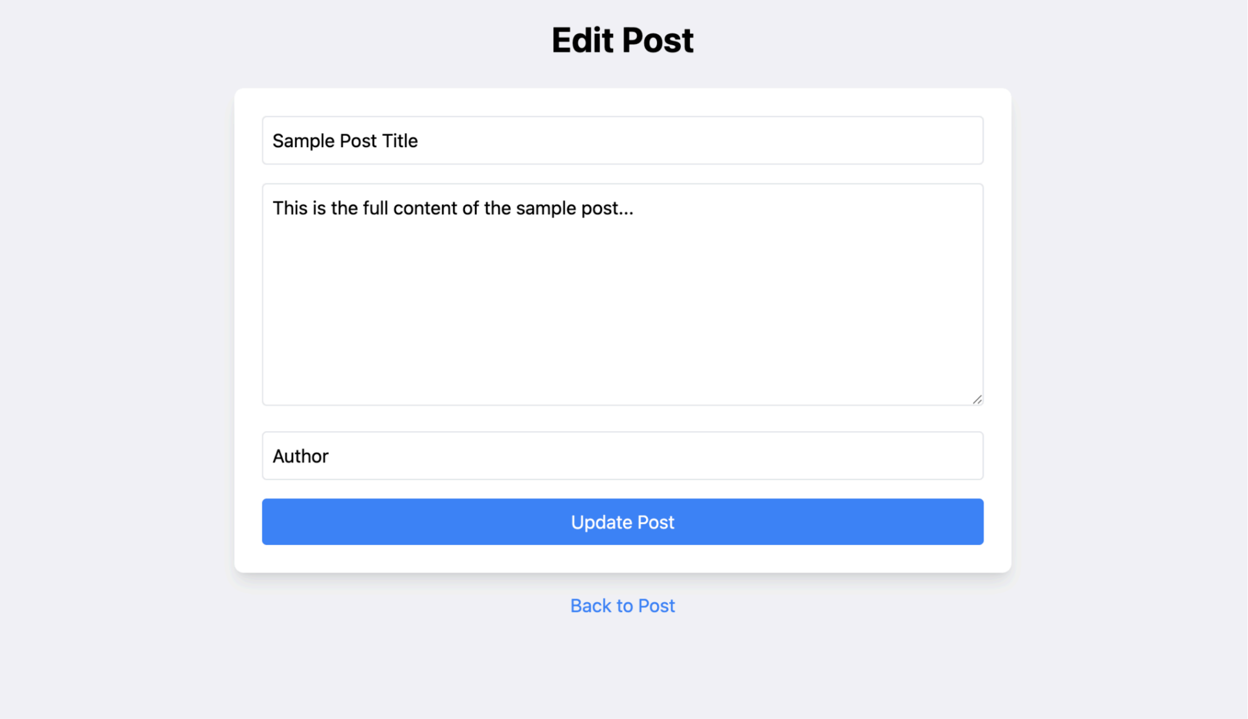
By: Author | Date: 2025-04-29



Sample Post Title

This is the full content of the sample post. It can be quite long and detailed, explaining the topic in depth.

By: Author | Date: 2025-04-29



Edit Post

Sample Post Title

This is the full content of the sample post...

Author