

# Mini social media app

---

## Task: Building a Social media app

### Objective:

Develop a Node.js web application using Express and EJS as the template engine. The application should perform CRUD operations, handle database relationships, and implement session-based authentication.

### Requirements:

#### 1. MVC Architecture:

- Structure your application following the MVC (Model-View-Controller) pattern.
- Create separate folders for models, views, and controllers.

#### 2. Session-Based Authentication:

- Implement user authentication .
- Users should be able to register, log in, and log out.
- Passwords must be hashed using bcrypt.
- Use middleware to protect routes that require authentication.

#### 3. Database Relationships:

- Use Sequelize as the ORM to interact with the database.
- Set up models with the following relationships:
  - **One-to-Many:** A `User` model should have many `Post` models.
  - **One-to-Many:** A `Post` model should have many `Comment` models.
  - **One-to-One:** Implement a `User` model with a `Profile` model where each user has one associated profile.

#### 4. EJS Template Engine:

- Use EJS as the template engine to render views.

- Create EJS templates for user registration, login, and displaying user-specific content (e.g., profile, posts, and comments).

## 5. CRUD Operations:

- Implement CRUD operations for `User`, `Post`, and `Comment` resources.
- Ensure that only authenticated users can create, update, or delete their own posts and comments.

## 6. Validation and Error Handling:

- Implement input validation to ensure data integrity (e.g., non-empty fields, valid email format).
- Handle errors gracefully and provide meaningful error messages to users.

## 7. Project Structure: ( HINT)

- Organize your project as follows:

```
project-root/
├── config/
│   └── database.js
├── controllers/
│   ├── authController.js
│   ├── userController.js
│   ├── postController.js
│   └── commentController.js
├── models/
│   ├── index.js
│   ├── user.js
│   ├── post.js
│   ├── comment.js
│   └── role.js
├── routes/
│   ├── authRoutes.js
│   ├── userRoutes.js
│   ├── postRoutes.js
│   └── commentRoutes.js
├── views/
│   └── partials/
```

```
|   |   └─ header.ejs
|   └─ auth/
|   |   └─ login.ejs
|   |   └─ register.ejs
|   └─ user/
|   |   └─ profile.ejs
|   |   └─ posts.ejs
|   └─ post/
|   |   └─ index.ejs
|   |   └─ detail.ejs
|   └─ index.ejs
└─ middlewares/
   └─ authMiddleware.js
└─ .env
└─ app.js
└─ package.json
```

**Submission:**

- Submit your project as a GitHub repository.

**Bonus:**

- Implement pagination for the `Post` and `Comment` views.
-