

1. **User registration:**
  - Create a route for handling user registration requests. This route will accept POST requests and validate the request data (email, phone, password, confirm password).
  - If the validation is successful, create a new user in the database with the provided data.
  - Return a success message to the client.
  
2. **Registered user can create, read, and update their profile:**
  - Create a route for handling profile updates. This route will accept PUT requests and validate the request data (first name, last name, role).
  - If the validation is successful, update the user's profile in the database with the provided data.
  - Return a success message to the client.
  
3. **User with writer role can create posts and read, update and delete only their posts:**
  - Create a route for handling post creation. This route will accept POST requests and validate the request data (title, post).
  - If the validation is successful, create a new post in the database with the provided data and associate it with the logged-in user.
  - Return a success message to the client.
  - Create routes for reading, updating, and deleting posts. These routes will accept GET, PUT, and DELETE requests, respectively, and retrieve or modify the relevant post from the database.
  
4. **User with writer role can read comments on his posts:**
  - Create a route for reading comments on a post. This route will accept GET requests and retrieve the comments for the specified post from the database.
5. **User with editor role can read all posts and comment on posts:**
  - Create a route for reading all posts. This route will accept GET requests and retrieve all posts from the database.
  - Create a route for creating comments on a post. This route will accept POST requests and validate the request data (comment).
  - If the validation is successful, create a new comment in the database with the provided data and associate it with the specified post.
  - Return a success message to the client.