Assignment

Objective

Build a small web application to manage user-generated blog posts. The app should support authentication (JWT), REST APIs, file upload, SQL relational logic, and include a basic UI. Swagger/OpenAPI documentation is optional but appreciated.

Requirements

1.Authentication

- Implement **JWT-based** login and registration.
- Only authenticated users can create/update/delete blog posts.
- `POST /auth/register` – Register a new user (email, password, name)
- `POST /auth/login` – Login with JWT token generation
- `GET /auth/profile` – Get logged-in user's profile (JWT required)

---

2. Blog Post CRUD REST API (Use Middleware to Authenticate)

Create RESTful APIs for:

- `POST /posts` - Create a blog post
- `GET /posts` - Get all blog posts
- `GET /posts/:id` - Get single blog post
- `PUT /posts/:id` - Update blog post
- `DELETE /posts/:id` - Delete blog post

Each post should have:

- Title (string)
- Content (text)
- Author (linked to user)
- Cover image (upload)
- Created & updated timestamps

3. File Upload

- Enable users to upload a **cover image** for each blog post.
- Store the image locally or on cloud storage (S3 preferred for bonus).
- Include the image URL in the response.

---

4. Database Design (SQL)

Use **MySQL** with proper normalization. Include at least:

Tables:

- `users`
- `posts` (foreign key to `users`)

> Use proper SQL joins in your API to return post details with user information.

Example query requirement:

- Get all posts along with the name/email of the author using a JOIN.

---

5. Basic UI (Frontend)

- Simple interface (HTML/CSS/JS, or React/Next.js) with:
  - Login/Registration
  - Create Post form
  - Post listing page
  - View single post with cover image

---

6. Swagger Documentation (Optional – Bonus Extra Point)

- Add Swagger/OpenAPI documentation for your API endpoints.
- Expose it via `/docs` route.

Deliverables

- Live url Frontend and Backend (Optional - Bonus Extra Point)
- Source code (GitHub repo or zipped project)
- SQL schema or migration files
- README including:
    - Project setup instructions
    - Technologies used
    - Optional: Swagger API usage guide
    - example.env

> **Time to Complete**: 3 days

Good luck, and we're excited to see what you build!