**WebOps Backend Dev Task — Final Implementation Summary**

**Features Implemented**

**Authentication (Login & Signup)**

* Implemented using JWT.
* Users have: name, email, password, image.
* Login & Signup endpoints fully implemented in AuthController.
* Other endpoints protected with before\_action :authorize\_request.

**CRUD for Posts**

* Fields: title, body, tags, author.
* Posts belong to a user as author (author\_id).
* Only authors can update/delete their own posts.
* Tags validation present (validates :tags, presence: true).
* Posts automatically deleted after 24 hours using Sidekiq job (DeleteOldPostsJob), scheduled every hour via Sidekiq-Cron.

**Comments on Posts**

* Users can add comments to any post.
* Only comment author can update or delete their own comment.
* Comment creation, update, and delete endpoints implemented and fully tested.

**Other Backend Setup**

* JWT-based auth used for all endpoints (except login/signup).
* Docker Compose used to run Rails app, Sidekiq, Redis, and Postgres.
* Sidekiq configured with Redis.
* Sidekiq-Cron configured to run deletion job every hour.

**Tests Summary**

**Auth Spec (auth\_spec.rb)**

**Happy scenarios:**

* Signup with valid parameters → returns 201 Created.
* Login with correct credentials → returns 200 OK.

**Unhappy scenarios:**

* Signup with missing parameters → returns 422 Unprocessable Entity.
* Login with invalid credentials → returns 401 Unauthorized.

**Posts Spec (posts\_spec.rb)**

**Happy scenarios:**

* Create post with valid params → returns 201 Created.
* Get all posts → returns 200 OK.
* Update post as author → returns 200 OK and updates title.
* Delete post as author → returns 204 No Content.

**Unhappy scenarios:**

* Create post without token → returns 401 Unauthorized.
* Update another user’s post → returns 403 Forbidden.
* Create post with missing fields → returns 422 Unprocessable Entity.

**Comments Spec (comments\_spec.rb)**

**Happy scenarios:**

* Create comment with valid params → returns 201 Created.
* Update comment as author → returns 200 OK.
* Delete comment as author → returns 204 No Content.

**Unhappy scenarios:**

* Create comment without token → returns 401 Unauthorized.
* Update or delete another user’s comment → returns 401 Unauthorized (guard clause present).