

Stage 1

Achieved:

- Entities Drafted
- ERD first Draft
- Plan what Documentation I can answer now
-

Stage 2

Achieved:

- Setup up Directory with all deliverables
- Answered Q1 to 4

Challenges: Not Confident in ERD

Moved to next Stage: ERD

Stage 3

Achieved:

- Mods Identified and installed
- setup.py and app.py setup for current functionality
- User Model created with minimal attributes
- Cli route established and tested successfully creates and seeds the DB
- User route (login and signup) created and tested with insomnia

Challenges:

- Was getting a privilege error, fixed with `"grant create on schema public to user;"`
- Imported ma to routes instead of models.users

Moved to next Stage:

- ERD final draft, could move to stage 5 as current ERD is more than detailed enough for current development
- Planning Questions, I wanted to code. I will re-access which questions should be answered at beginning of stage 4.

Goals Dropped:

- No planned features have currently be dropped.

New Possible Features:

- I will do some light research of redirecting routes for my homepage
- After logging in just return JWT and email, the rest seems redundant

Stage 4

Achieved:

- ERD finalised will review on last stage to ensure it aligns with the project
- Look up 1 user
- Added token requirement so user must successfully login to interact with the webserver
- Excluded email and password for security
- Log CRUD setup
- Wrote a function that returns all user's logs. Useful for Read and Delete.
- Made back-populating relationship between users and logs, then added parameters to schema returns to remove any circular errors
- Added Testing comments into Stage overview

Challenges:

- I had an issue with loading all a single users logs. Once I identified that the problem was that the user had multiple logs I used `db.session.query` so could iterate outputs, supporting multiple logs.

Testing:

- Ran through each endpoint in Insomnia, All endpoint functioned successfully
- While testing the Log delete route, I found I could delete any log as any user. My first task in Stage 5 will be to ensure that logs can only be deleted by their owners.

Moved to next Stage:

- Password Hashing
- Planning Documents (can be finished in final stage)

Goals Dropped:

- Redirect dropped as the method only supported GET
- Considered adding admin tiers, however decided to stay with binary Booleans.

New Possible Features:

- Add public bool so users can choose not to be looked up/ logs
- A tier of admin for different degrees of access

Stage 5

Achieved:

- Made Auth.py which contains an authorisation function to ensure the active user only has authorisation over their content (except admin which has all access) .
- Created a comment model and route, then backpopulated with users and Logs.
- Recipe Model made, No route as it only exists as a nest off a recipe.
- Made a Clone route to copy existing logs to a users DB, I did not create a model for clone as it would use Recipe and Log models.
- Hashed password with bcrypt for safe storage and added password to the login process.

Challenges:

- The clone route presented issues the follow issues;
 - My initial attempt would move the log and its contents to the user actioning the route, instead of creating a new instance of the log.
 - After resolving I could copy the log but the recipe was taken from the original log.
 - As suggested by the educator I iterated the recipe after the log commit.

Testing:

- Clone route tested with Insomnia and cli command
 - **Cli Command (Flask db test)** mimicked parse and resolved in a print. This Helped me Identify what that parse contained and how to select key parts.
 - **Insomnia** as a final test to review API function.

Moved to next Stage:

- Nothing moved to next stage

Goals Dropped:

- Creating a public bool dropped as web server is for the sharing of logs

New Possible Features:

- The clone route will be reviewed, it works correctly but code could be minimised.

Stage 6

Achieved:

- Assessment complete, target function achieved.

Challenges:

- Trying to make a loop for the recipes nested into the log, clone blueprint.

Testing:

- Printed out all route endpoints and made a checklist from each endpoint. Used Insomnia and assessed return to ensure logical function.

Moved to next Stage:

- Finishing documentation.