

Attendees: Sonia Singh, Eduardo Bermudez, Xanthippe Pentovelis, Hoai Kiet Tran, Omar Eshatti

Notes

- As a recently formed team, students have created their Github repository.
- Students decided to use the programming language JavaScript for project execution.
- For the backend students chose the running environment known as [Node.js](#)

User stories:

(US-1) User Registration

Tasks (Issues)

1. Create registration page UI (form + basic styling)
2. Client-side validation (email format, password rules)
3. Backend endpoint: POST /auth/register
4. Hash passwords + store user in DB
5. Handle errors (duplicate email, invalid input)
6. Success flow (redirect + success message)
7. Quick test: register works end-to-end

(US-2) User Login - Eduardo Bermudez

As a returning user,
I want to log in with my credentials,
so that I can access my account.

Acceptance Criteria

- Login form takes email + password

- Invalid credentials show error message
- Successful login creates session / token
- User lands on a protected page after login

Tasks (Issues)

1. Create login page UI
2. Backend endpoint: POST /auth/login
3. Verify password (compare hash)
4. Create auth session (JWT or cookie session)
5. Store token / session on frontend
6. Redirect after login
7. Quick test: login works end-to-end

(US-3) Logout - Omar Eshatti

As a logged-in user,
I want to log out,
so that my account is not accessible on this device.

Acceptance Criteria

- Logout button exists
- Session/token is cleared
- User is redirected to login/home

Tasks (Issues)

1. Add logout button to navbar/menu
2. Frontend: clear token/session
3. (Optional) Backend logout endpoint if using cookies
4. Redirect to login/home
5. Quick test: protected page blocked after logout

(US-4) Authentication State + Protected Routes - Eduardo Bermudez

As a user,

I want pages to be protected unless I'm logged in,
so that only authenticated users can access account features.

Acceptance Criteria

- If not logged in → trying to access protected pages redirects to login
- If logged in → user can access protected pages
- Auth state persists on refresh (if using token/cookie)

Tasks (Issues)

1. Define which routes/pages are protected (profile, meal planner later, etc.)
2. Add auth guard (frontend route protection)
3. Backend middleware to protect API routes
4. Persist auth state (read token/cookie on app load)
5. Quick test: refresh keeps user logged in

(US-5) View Profile - Xan

As a logged-in user,

I want to view my profile,
so that I can see my diet preferences and allergies.

Acceptance Criteria

- Profile page shows user info (email optional)
- Shows diet preferences and allergies fields

- Loads data from DB for the logged-in user only

Tasks (Issues)

1. Create profile page UI
2. Backend endpoint: GET /users/me
3. Connect frontend to fetch and display profile data
4. Ensure auth required (only logged-in users)
5. Quick test: correct user data displays

(US-6) Edit Profile (Diet Preferences + Allergies) - Tran

As a logged-in user,

I want to update my diet preferences and allergies,
so that recipes and meal plans can match my needs.

Acceptance Criteria

- User can edit diet preference (dropdown/multi-select)
- User can add/remove allergies (list/tags)
- Save updates in DB
- Validation + confirmation message

Tasks (Issues)

1. Add editable fields on profile page (diet + allergies)
2. Backend endpoint: PUT /users/me
3. Validate inputs (no empty allergy entries, etc.)
4. Save changes to DB
5. Show “saved” feedback to user
6. Quick test: updates persist after refresh

Action items

- Github page

- Create user stories
- Create tasks
- Make readme file
- Create sprint planning table in excel
- Sprint planning table: Xan and Kiet by Feb 2
- Document log
-