Sprint 3: Planning Document -

Roles:

👥 Sprint 3 Team Roles (5 Members)

🧩 1️⃣ DevOps & Environment Lead

Main Responsibility: Make sure the project runs for everyone.

Tasks:

- Create `docker-compose.yml`
- Configure:
    - Node container
    - MySQL container
- Set up environment variables
- Connect Express to MySQL
- Write setup instructions in README
- Help others troubleshoot Docker issues

Code Contribution:

- Database connection logic
- Environment config files
- Possibly middleware setup

🎯 Goal:
Everyone can run:

```
docker-compose up
```

without errors.

🗄 2️⃣ Database & Models Lead

Main Responsibility: Design and implement database structure.

Tasks:

- Design tables:

    - Users
    - Listings
    - Tags
    - Listing_Tags (if needed)
- Create SQL schema
- Add seed data
- Create ER diagram (for PDF)
- Write model/query functions:
    - getAllUsers()
    - getUserById()
    - getAllListings()
    - getListingById()
    - getTags()

Code Contribution:

- SQL files
- Model files (e.g., userModel.js, listingModel.js)

🎯 Goal:
All pages pull real data from MySQL.

👤 3️⃣ Users Feature Developer

Main Responsibility: Implement user-related pages.

Pages Required:

- Users list page
- User profile page

Tasks:

- Create routes:
    - `/users`
    - `/users/:id`
- Connect routes to model functions
- Create Pug templates:
    - users.pug
    - user-profile.pug
- Ensure dynamic rendering (no hardcoded data)

🎯 Goal:
Users page and profile page fully working with DB data.

📦 4️⃣ Listings Feature Developer

Main Responsibility: Implement listings features.

Pages Required:

- Listings page
- Listing detail page

Tasks:

- Create routes:
    - `/listings`
    - `/listings/:id`
- Connect to database queries
- Create:
    - listings.pug
    - listing-detail.pug
- Ensure relationship with users (e.g., listing owner)

🎯 Goal:
Listings and detail pages working and connected to DB.

🏷️ 5️⃣ Tags + QA + Documentation Lead

Main Responsibility: Categories/tags + quality control + submission preparation.

Pages Required:

- Tags/categories page
- Filtering by tag (if implemented)

Technical Tasks:

- Route:
    - `/tags`
    - `/tags/:id` (optional)
- Create tags.pug

- Connect listings to tags

QA Tasks:

- Test all routes
- Ensure no hardcoded data
- Check foreign key relationships
- Fix minor bugs

Documentation Tasks:

- Collect:
  - User stories
  - GitHub metrics screenshot
  - Kanban board screenshot
  - Meeting records
- Prepare final PDF

🎯 Goal:
Everything works AND submission requirements are complete.

Timeline Planning:

☑ Sprint 3 Timeline (7 Days, 5 Members)

👥 Team Roles (Recommended)

- **Dev 1** → Docker & Environment Lead
- **Dev 2** → Database & Schema Lead
- **Dev 3** → Users Feature
- **Dev 4** → Listings Feature
- **Dev 5** → Tags + QA + Documentation

Everyone must still commit code.

📅 Day-by-Day Plan

Day 1 – Planning & Database Design (All Members)

Goal: No coding until structure is clear.

Tasks:

- Confirm user stories being implemented
- Design database tables together:
  - Users
  - Listings
  - Tags
  - Listing_Tags (if needed)
- Draw ER diagram
- Create GitHub issues
- Set up Kanban board

Output:

- Approved DB design
- Task allocation confirmed
- GitHub project board created

Day 2 – Database + Docker Setup

Dev 2

- Write SQL schema
- Add seed data
- Test locally

Dev 1

- Create Docker setup:
  - MySQL container
  - Node container
- Connect Express to MySQL

Everyone

- Pull repo
- Run:

```
docker-compose up
```

- Confirm it works

⚠️ Everyone must prove they can run it.

▨ Day 3 – Backend Models (Parallel Work Begins)

Now people can split.

👤 Dev 2

- Create model functions:
    - getAllUsers()
    - getUserById()
    - getAllListings()
    - getListingById()
    - getTags()

👤 Dev 3

- Start Users routes

👤 Dev 4

- Start Listings routes

👤 Dev 5

- Start Tags route
- Help test queries

▨ Day 4 – Express Routes Working

Goal: Routes return real database data.

By end of Day 4:

- `/users` works
- `/users/:id` works
- `/listings` works
- `/listings/:id` works
- `/tags` works

Even if just returning JSON at this stage.

Day 5 – Pug Templates

Now build UI.

Dev 3

- users.pug
- user-profile.pug

Dev 4

- listings.pug
- listing-detail.pug

Dev 5

- tags page
- Styling consistency

Dev 1

- Fix Docker bugs
- Clean config
- Help integration

Day 6 – Integration & Testing

All members:

- Remove hardcoded data
- Fix query issues
- Test:
  - Foreign key relationships
  - Page navigation
  - Broken routes
- Ensure:
  - No one’s branch is unmerged
  - All pages use database

Dev 5:

- Start compiling PDF content

- Collect screenshots:
    - GitHub metrics
    - Kanban board
    - Commits

▨ Day 7 – Final Polish & Submission Prep

Technical:

- Final bug fixes
- Clean code
- README updated
- Docker tested from scratch

Documentation:

Include in PDF:

- User stories
- Database design diagram
- Task allocation breakdown
- GitHub repo link
- GitHub project link
- Metrics screenshot
- Kanban screenshot
- Meeting records