

SWOPLET

MASTER TEAM PROJECT SOSE 2025 MILESTONE 2: MORE DETAILED REQUIREMENTS, SPECS, ARCHITECTURE, UI MOCK-UPS AND VERTICAL SW PROTOTYPE

Team Project / GDSD Summer 2025 Team 6

Team Lead Email: muhammad-hamza.butt@informatik.hs-fulda.de

TEAM MEMBERS

Team Member	Role
Hamza Butt	Team Lead
Karan Patel	Github Master
Pooja Puthu Vayal	Backend Lead
Akhil Sajan	Frontend Lead
Hammad Asif	Frontend Developer

HISTORY

Date	Version	Notes
29/05/2025	First Draft	Initial Submission

1. EXECUTIVE SUMMERY

University students, teachers, and employees in this competitive era of time frequently struggle to buy, sell, and exchange goods within their educational community. The existing platforms are mostly public-oriented, and this can cause problems in security, authenticity, and relevance. In addition, users get lost in setting fair prices for products, resulting in poor bargaining and missed opportunities. For Fulda University student association these concerns delay the capability of having a credible, available, and efficient marketplace that meets their specific needs.

To address these challenges, Fulda University Masters team is developing a focused web-based site. The site will be restricted to serving the Fulda University exclusively, allowing students, faculty members, and staff to list things for sale securely, search ads, and complete transactions. Key features include:

- Listing items for sale with complete descriptions and photos.
- Instant searching and browsing of items available for easy selection through basic filters.
- Restriction of user interactions to members of only the involved university communities through a secure messaging system.
- Admin moderation to enable posts to go live once approved and for removing offending content or users, ensuring quality and safety.

Besides the basic features, the platform is built in a flexible way that allows it to continuously evolve and add new, creative, and top-quality features.

Among the platform's unique and competitive features is the integration of an Artificial Intelligence (AI) model that analyzes market data to provide buyers with suggestions on the average market price of similar products. This feature empowers users with valuable pricing insights, enabling sellers and buyers to negotiate more wisely and more fairly. Through the use of AI-driven price estimating, the platform not only enhances transaction transparency but also gets a significant advantage over existing marketplaces, acquiring and maintaining users through the provision of tangible value to the shopping experience.

The application is developed by a dynamic group of about five students who closely collaborate under the instructions of the CTO. The diverse background of the team and Agile Systems Engineering approach enable them to iteratively evolve the functionality of the platform to maximize the fulfillment of user needs. With careful milestones, the team is endeavoring to continuously monitor and enhance the service to offer robustness, security, and user satisfaction. Lastly, the objective is to develop a highly respected, vibrant online marketplace dedicated to the Fulda University community and providing superior value while building a positive campus commerce environment.

2. FUNCTIONAL REQUIREMENTS ALONG WITH PRIORITY

2.1 PRIORITY 1 - MUST HAVE FUNCTIONAL REQUIREMENTS

2.1.1 All Users

- **FR1 – User Registration & Affiliation Check:** Users must register using a verified institutional email (e.g., @hs-fulda.de or @sfsu.edu). This ensures that only affiliated members can access the platform.
- **FR6 – Browse and Search Listings:** Users should be able to explore available items/services and use search functionality with keywords to locate specific listings.
- **FR7 – Filter and Sort Listings:** Users can narrow down results using filters (e.g., category, price, tags) and sort by parameters like newest, oldest, or price (low-high, high-low).

2.1.2 Buyer

- **FR8 – Internal Messaging:** Enables buyers to communicate securely with sellers through a built-in messaging system, without sharing personal contact details.
- **FR9 – Wish listing a Product:** Buyer can browse and wish list a product for future use.

2.1.3 Seller

- **FR3 – Post New Item/Service:** Authenticated sellers can create listings by entering essential details like title, description, category, price, and contact preferences.
- **FR4 – Upload Media for Listing:** Sellers can add multiple images or videos to visually showcase the product or service they are offering.
- **FR8 – Internal Messaging:** Sellers can respond to buyer inquiries through the platform's internal messaging system.
- **FR10 - Seller Dashboard:** A dedicated dashboard allows sellers to view and manage their listings (edit, delete, pause, mark sold) and track engagement.
- **FR20 - Temporarily Hide Listing:** Sellers can temporarily pause a listing from public view without deleting it; useful for restocking or updating
- **FR5 - Categorize Listings:** Sellers must select appropriate categories and add relevant tags to ensure their listings are easily discoverable

2.1.4 Admin

- **FR11 – Admin Approval for Listings:** Every listing must be reviewed and approved by a moderator/admin before becoming publicly visible, ensuring quality and safety.
- **FR12 – Admin Moderation Capabilities:** Moderators can remove inappropriate content, block or ban abusive users, and maintain platform standards.

2.2 PRIORITY 2 - DESIRED FUNCTIONAL REQUIREMENTS

2.2.1 Buyer

- **FR2 - User Profile Management:** Users should be able to view, edit, and update their profile details, including name, photo, and short bio. •
FR26 – Rate Seller: Buyer can post a rating for seller
- **FR13 - Report Inappropriate Listings:** Users can flag inappropriate listings/ messages for admin review.
- **FR24 – AI-Powered Market Price Estimation :** The system shall integrate an AI model that analyzes historical and real-time market data of similar product listings to provide users with an estimated average market price.

2.2.2 Seller

- **FR19 – Highlight Discounted Listings:** Listings with reduced prices are automatically added to a “Price Drops” section for greater visibility and buyer engagement.
- **FR2 - User Profile Management:** Users should be able to view, edit, and update their profile details, including name, photo, and short bio.

2.3 PRIORITY 3 - OPPORTUNISTIC FUNCTIONAL REQUIREMENTS

2.3.1 Users (Seller, Buyer):

- **FR17 - Accept Terms and Privacy Policy:** Users must accept the Terms of Use and Privacy Policy during account creation. •
FR16 - Share Listings Externally: Users can copy or share the listing link via social platforms or email.

- **FR21 - Password Reset Flow:** Users can reset forgotten passwords through an email-based recovery system.
- **FR23 - FAQ/Help Page:** A static help page with answers to common questions (e.g., how to post, flag, contact seller).

2.3.2 Buyer:

- **FR18 - Track Price History:** Shows a small log of previous prices to help buyers see if the price has dropped.

2.3.3 Seller:

- **FR14 - Flash Sale Listing:** Sellers can mark items as “Flash Sale” for urgent attention (e.g., moving out, quick deals).
- **FR15 - Automatic Listing Expiry:** Listings automatically expire after a set period (e.g., 30 days), with the option to renew.

2.3.4 Moderator:

- **FR13 - Report Inappropriate Listings:** Users can flag inappropriate listings/messages for admin review (admin will be responsible for reviewing the flagged content).
- **FR22 - Contact/Support Form:** A contact form routes inquiries or issues to a monitored support inbox.

3. LIST OF MAIN DATA ITEMS AND ENTITIES

3.1 USER TYPES AND PRIVILEGES

User Type	Description	Privileges
Guest	Unregistered visitor browsing the platform	<ul style="list-style-type: none"> • View public listings • Apply filters to search products
Buyer		<ul style="list-style-type: none"> • Register and log in

User Type	Description	Privileges
-----------	-------------	------------

	Verified Fulda user who buys products	<ul style="list-style-type: none"> • Search and filter products • Contact sellers via in-site messaging • Add products to favorites • Compare product prices • Report listings or sellers
Seller	Verified Fulda user posting products for sale	<ul style="list-style-type: none"> • Register and log in • Create, update, delete product listings • Upload media (images) • Respond to buyer messages • View listings • Set time limits for product availability • Report buyers
Moderator	System user with full access and moderation privileges	<ul style="list-style-type: none"> • Approve or reject listings • Manage flagged reports • Monitor messages for violations • Warn/ban users • Review and publish listing expiration (time limit) actions

3.2 MAIN DATA ENTITIES AND DESCRIPTIONS

Entity Name	Description
User	A registered individual on the platform (Guest, Buyer, Seller, Moderator)
Product Listing	Represents an item for sale. Includes title, description, price, condition, images, and time limits.
Media File	Images uploaded by sellers and reviewed by moderators.
Message	In-app chat communication between buyer and seller.
Favorite	Product saved by a buyer for later access and comparison.
Report	A complaint raised against a listing or user. Routed to moderators.
Search History	User-specific log of search queries and filters applied.

Entity Name	Description
Price Comparison	Dynamic table for comparing similar products by price, condition, and seller.
Time Limit	Optional setting by sellers to define listing expiration.

3.3 LOGICAL DATA STRUCTURE (HIGH-LEVEL OVERVIEW)

Users Table

Field Name	Description
user_id	Primary Key, unique user identifier
username	Display name of the user
password	Hashed user password
email	University email (@hs-fulda.de), unique
role	Refers to user role (Guest, Buyer, Seller, Moderator)
permission	Optional access control flag
profile_picture	URL to profile image
created_at	Account creation timestamp
modified_at	Last profile update timestamp
Is active	Boolean to store active status
warn_count	How many times moderator have warned

Product Listings Table

Field Name	Description
product_id	Primary Key, unique listing identifier
title	Title of the product
description	Detailed product information

Field Name	Description
category	Product category (e.g., Books, Electronics)
price	Listed price in €
condition	New or Used
tags	Optional keywords for search
location	Seller location
Created_by_id	Foreign Key referencing Users
status	Listing state (e.g., pending, approved)
created_at	Listing creation timestamp
updated_at	Last listing update timestamp

Media Files Table

Field Name	Description
Media id	Primary Key, unique media file ID
Product id	Foreign Key referencing Product Listings
file_path	Path or URL of the uploaded media
is_approved	Boolean flag for admin approval
uploaded_at	Media upload timestamp

Chat Table

Field Name	Description
chat id	Primary Key, unique chatID
product_id	Foreign Key referencing Product
product_owner_id	Foreign Key referencing Users

Field Name	Description
last_message_id	Foreign Key referencing Messages
created at	Chat timestamp

Messages Table

Field Name	Description
message_id	Primary Key, unique message ID
sender id	Foreign Key referencing Users
receiver id	Foreign Key referencing Users
product_id	Foreign Key referencing Product Listings
content	Message text
created at	Message sent timestamp

Favorites Table

Field Name	Description
favorite id	Primary Key
user id	Foreign Key referencing Users
product_id	Foreign Key referencing Product Listings
saved at	Timestamp when listing was bookmarked

Reports Table

Field Name	Description
report_id	Primary Key
reporter_id	Foreign Key referencing Users
targetType	Reported type (user or listing)

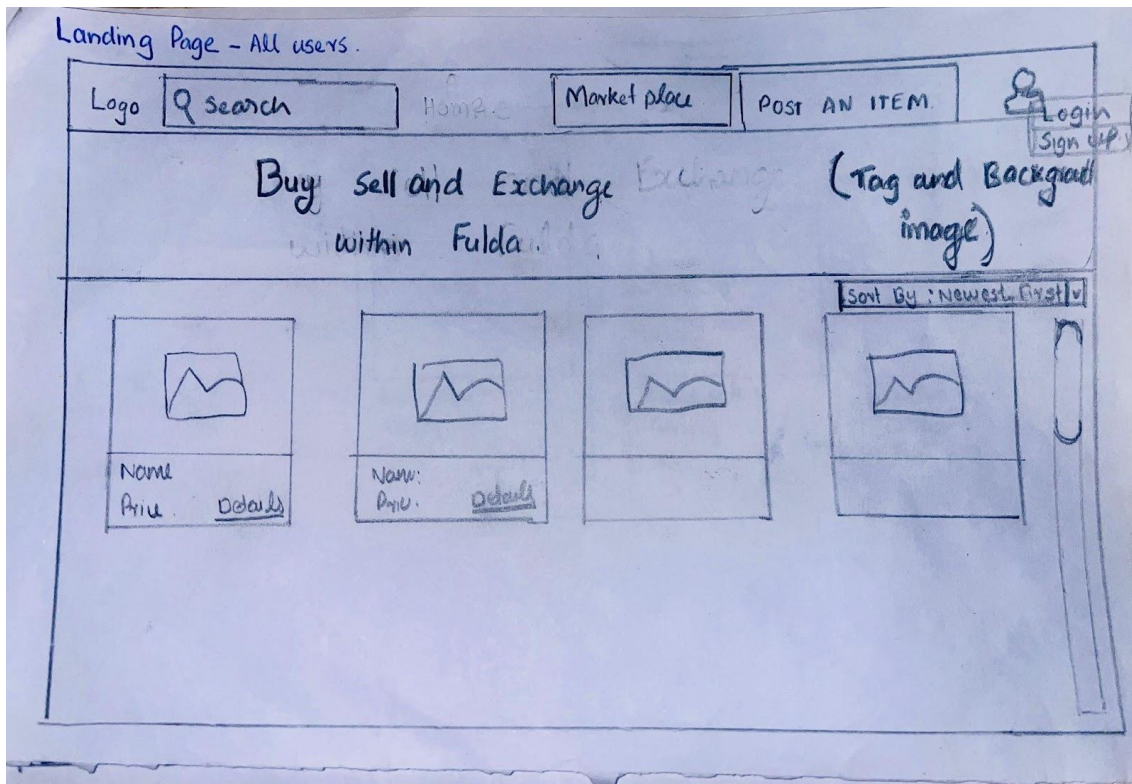
Field Name	Description
target_id	ID of the reported entity
reason	Reason for reporting
status	Report status (pending, resolved)
created at	Report submission timestamp

Search History Table

Field Name	Description
search id	Primary Key
user id	Foreign Key referencing Users
filters used	Search filters applied (stored as text/JSON)
timestamp	Time of search

4. NEW: UI MOCKUPS AND STORYBOARDS (HIGH LEVEL ONLY)

4.1 Landing Page



Description:

This is the landing page of our marketplace website, designed with a clean and engaging interface to welcome visitors. Users can browse featured product listings or choose to register and sign in as members. Once logged in, they gain full access to buy and sell items within the Fulda University community.

Key Features:

- **Main Actions:** Prominent "Browse" and "Sell" buttons for quick access to key functions. Browse button goes to marketplace and Post an Item button goes to login page. Details link of each product will lead to product detail page. But to contact a seller or add to wishlist option user needs to login.
- **Featured Listings:** A visually appealing horizontal carousel showcasing selected items (e.g., textbooks, electronics).

- **Login Section:** Streamlined login fields for returning users to sign in using their email and password.

4.2 User Login

The sketch shows a login page with the following elements:

- Header:** A horizontal bar containing a "Logo" on the left and "Home | About us" on the right.
- Form Fields:**
 - Email*:** A text input field with a required field asterisk.
 - Password*:** A text input field with a required field asterisk. To its right is a "Show" button for toggling password visibility.
 - Remember me:** A checkbox followed by the text "Remember me."
- Buttons:**
 - Sign in:** A button located below the "Remember me" checkbox.
 - Sign up:** A button located below the "Sign in" button, preceded by the text "New user?"

Description:

This login interface allows existing users to access their accounts on the Fulda University marketplace platform. The form provides a simple and secure way for users to authenticate and access personalized features.

Key Features:

- **User Authentication:** Email and password input fields for user authentication.
- **Remember me:** Checkbox for convenient future logins.

4.3 Registration Page

The wireframe shows a registration form titled "Sign-up (Registration Page)". At the top left is a "Logo" placeholder, and at the top right are "Home" and "About us" links. The form contains four input fields stacked vertically: "Full Name", "Email", "Password", and "Repeat Password". To the right of these fields is a box labeled "upload profile photo". Below the input fields are two buttons: "Register" and "Login". The "Login" button is preceded by the text "Already a user?".

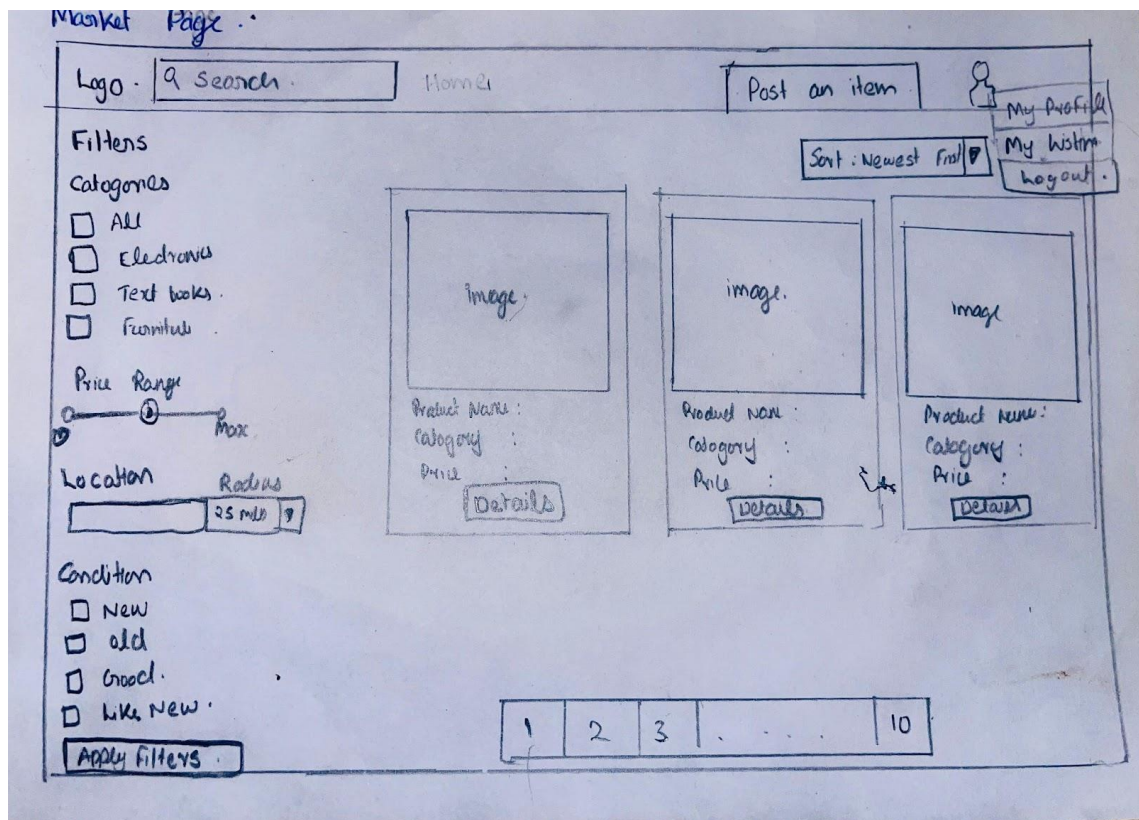
Description:

This page allows new users to sign up and become a part of the Fulda University marketplace. The form collects essential information such as full name, email, and password, and includes an image upload option to personalize user profiles.

Key Features:

- **User Information Fields:** Collects full name, email address, and password.
- **Image Upload:** Allows users to add a profile image during sign-up.

4.4 Marketplace



Description:

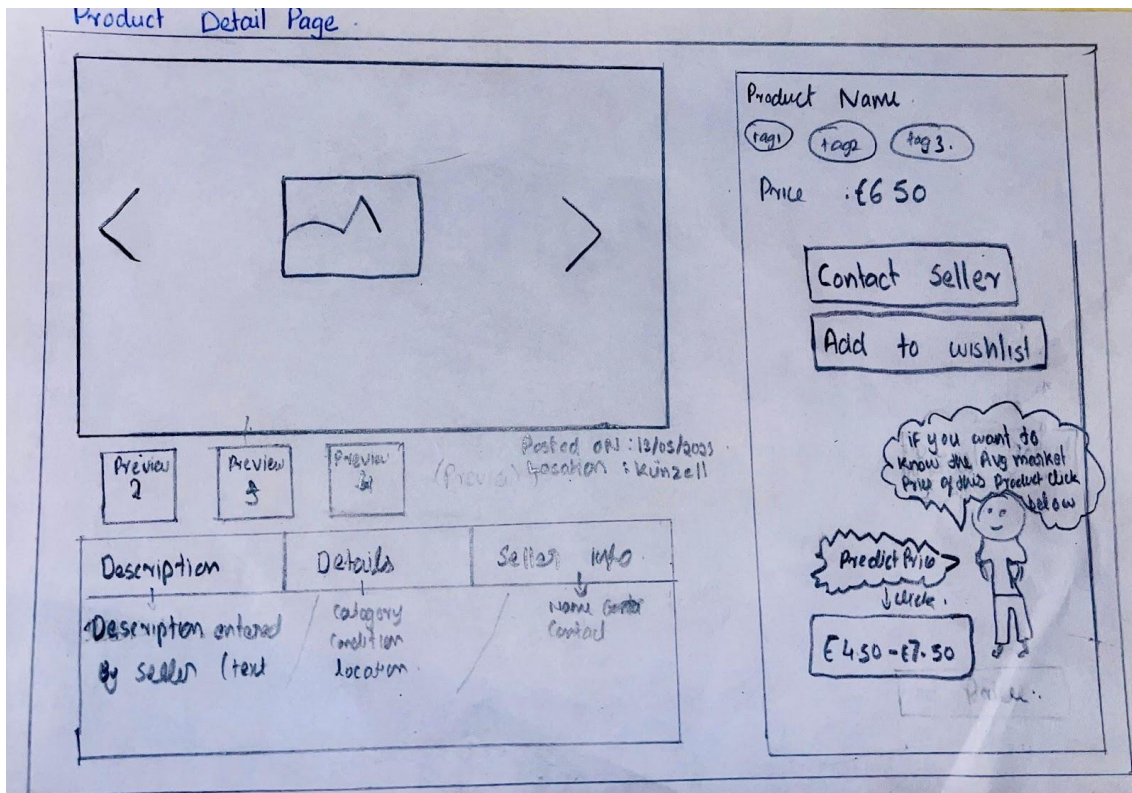
This is the main product browsing interface where users can explore items listed by other students. The marketplace is equipped with filtering and sorting features to help users quickly find relevant items across categories, price ranges, conditions and location.

Key Features:

- **Search and Filter Tools:** Search bar, category filters (e.g., Electronics, Textbooks), price range selectors, and condition checkboxes (New, Used, etc.).
- **Sorting Options:** Sort items by newest first or price drops.
- **Product Cards:** Each card includes an image, product name, category, price, and product tags.

4.5 Product Detail Page

Description:



This detailed product view provides comprehensive information about individual marketplace items. The page is designed to give potential buyers all the information they need to make informed purchasing decisions.

Key Features:

- **Product Image:** Large product image gallery with navigation arrows and thumbnail selection.
- **Product Information:** Product information including title, price, and descriptive tags, Pickup location and campus information.
- **Seller Details:** Seller contact information and rating system
- **Action buttons:** Contact Seller, Add to Wishlist
- **AI model to predict the average market price** of the product which helps buyer to negotiate.

4.6 Create Product Listing

The wireframe shows a form titled 'CREATE LISTING PAGE / Edit listing'. At the top, there is a header area with a 'Logo' placeholder, a search bar, and a user profile icon. The main form area contains the following fields and elements:

- Product Name/Title:** A single-line text input field.
- Description:** A large multi-line text area.
- Category:** A dropdown menu with 'Select' as the current value and a downward arrow.
- Condition:** A dropdown menu with 'select' as the current value and a downward arrow.
- Price:** A single-line text input field.
- Location:** A single-line text input field with a dropdown arrow.
- Photo Upload:** Four square buttons, each containing a camera icon, for uploading images. To the right of these buttons is the text '(upto 4)'.
- POST:** A large rectangular button at the bottom right of the form.

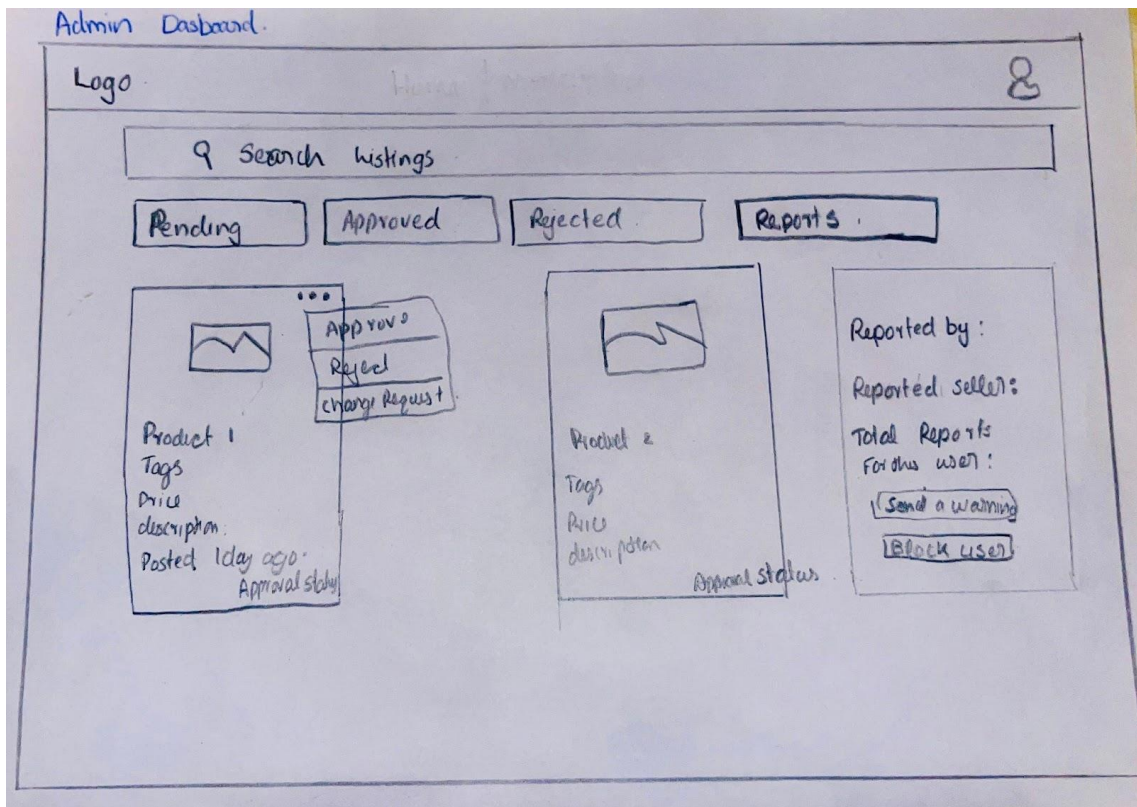
Description:

This comprehensive listing creation form allows users to add new items to the marketplace. The page provides all necessary fields and options for users to create detailed, attractive listings that will help them sell their items effectively.

Key Features:

- **Product Title:** Title input field for listing name
- **Product Description:** Large description text area for detailed item information
- **Category & Condition:** Category and condition dropdown selectors for proper classification.
- **Photo Upload and Preview:** Photo upload functionality supporting up to 4 images with upload indicators

4.7 Admin Dashboard



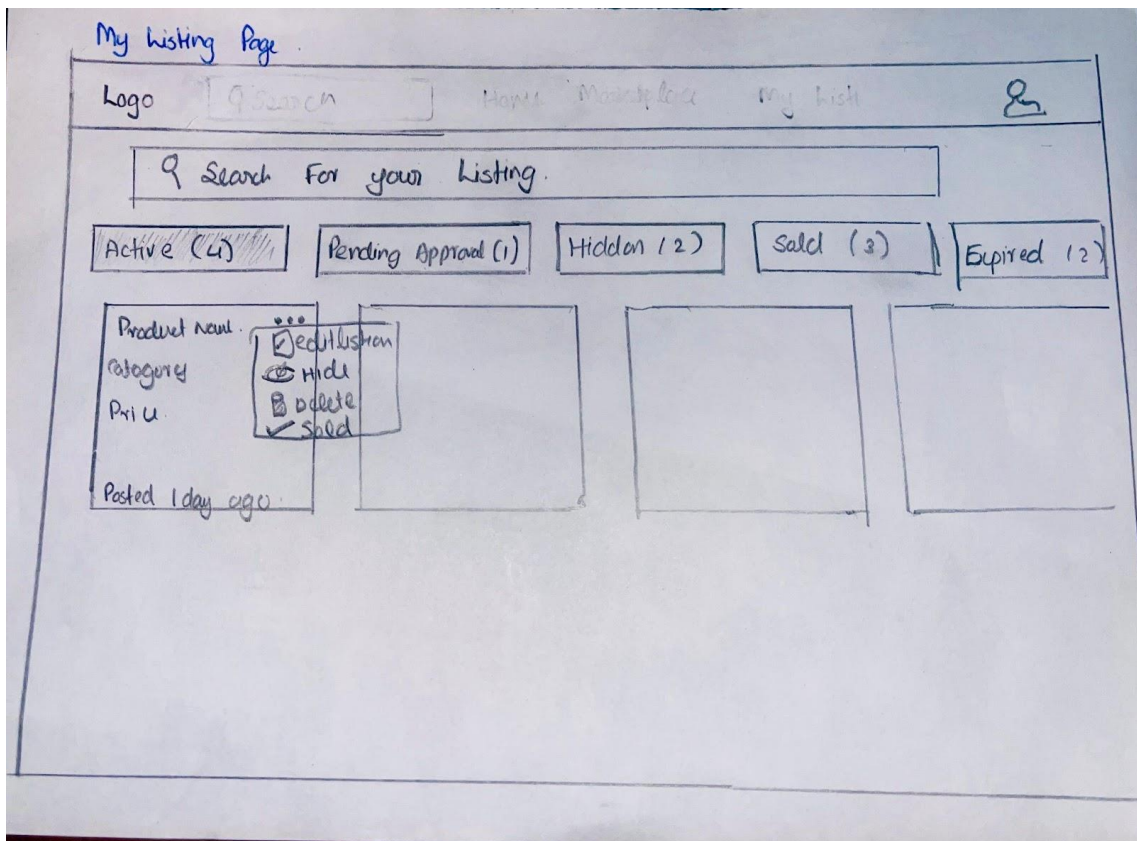
Description:

This administrative interface provides platform moderators with tools to oversee marketplace activities and manage user-submitted content. The dashboard enables efficient content moderation and platform management.

Key Features:

- **Status management tabs:** Approved, Pending, Rejected
- **Content Moderation:** Product listing cards with moderation controls
- **Action:** Approval workflow with multiple action options (Approve, Reject, Change, Request modifications)
- **Manage Reported Items:** Administrative actions (Delete, Dismiss, Report) for content management.

4.8 My Listings Page



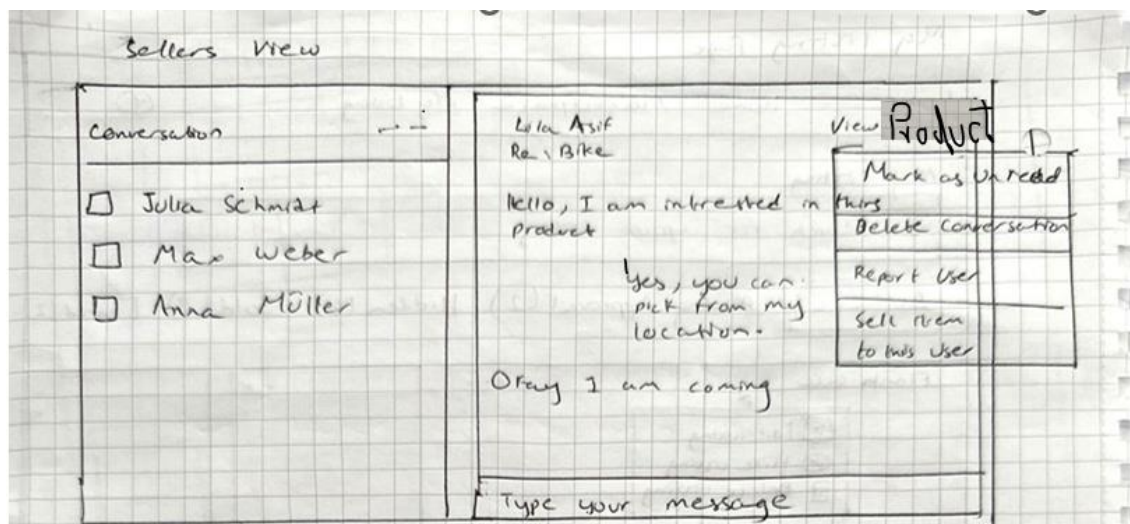
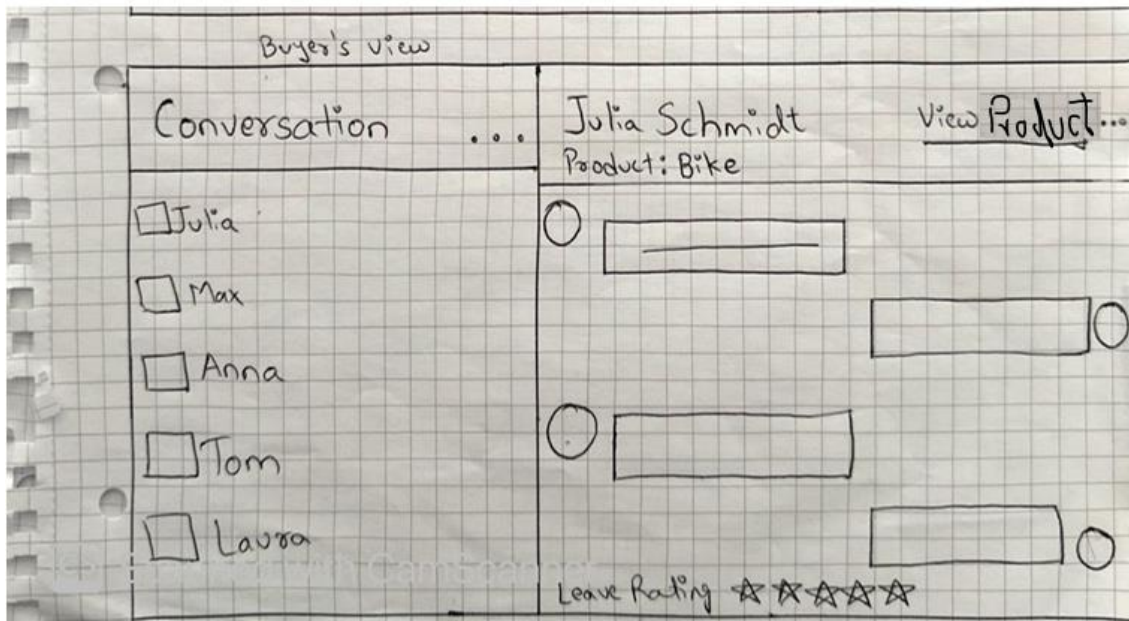
Description:

This page provides registered users with a personal dashboard to manage all their listings. It helps users keep track of the status of each product they have posted, whether it's active, pending approval, hidden, sold, or expired. Sellers can take quick actions like editing or deleting listings and monitor user engagement.

Key Features:

- **Search Bar:** Easily locate specific listings.
- **Status Tabs:** Filter listings based on status (Active, Pending, Hidden, Sold, Expired).
- **Product Cards:** Display product details along with options to edit, hide, or delete the listing.

4.9 Messaging Interface



Description:

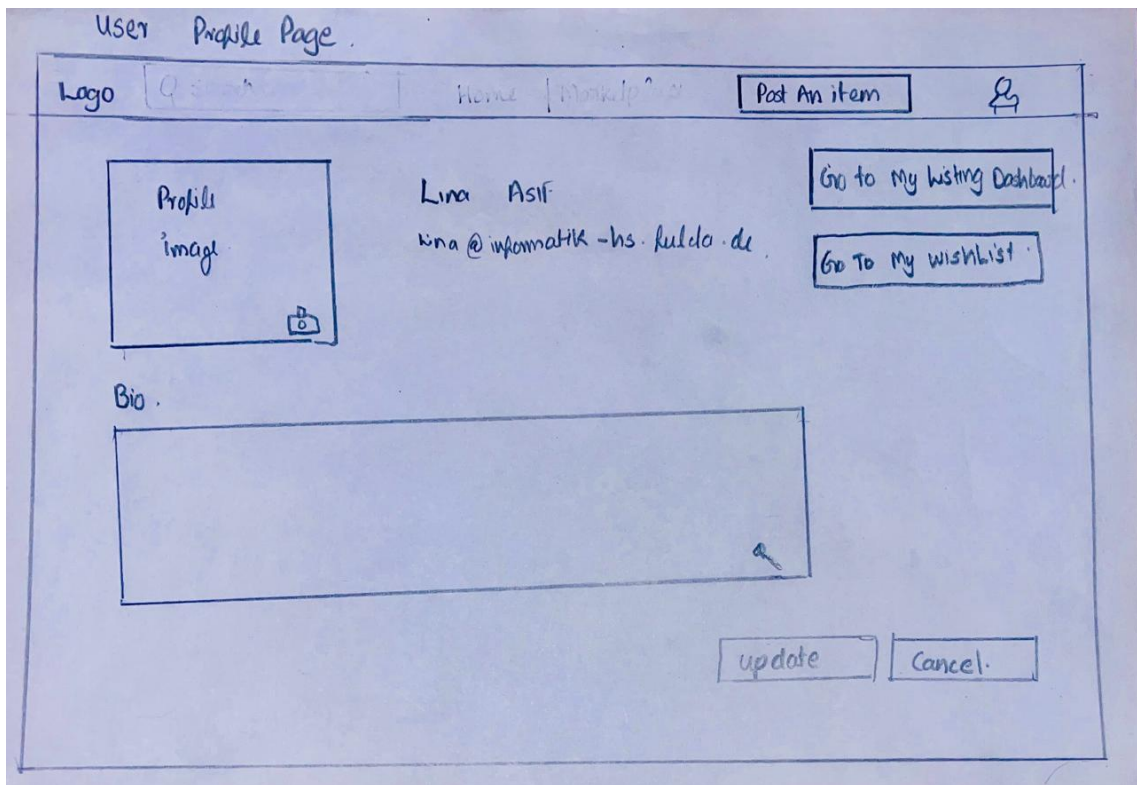
The messaging system allows buyers and sellers to interact with each other. Sellers can view a list of ongoing conversations, reply to messages, and take actions such as reporting users or marking conversations as unread. This feature ensures secure and organized communication within the platform.

Key Features:

- **Conversation List:** Sidebar with active message threads.
- **Message Panel:** Displays ongoing chat with buyers.
- **Action Menu:** Options to mark messages, report users, delete conversations, or finalize sales directly from the chat window.

- **Reply Box:** Easy-to-use input field for responding to messages.

4.10 User Profile



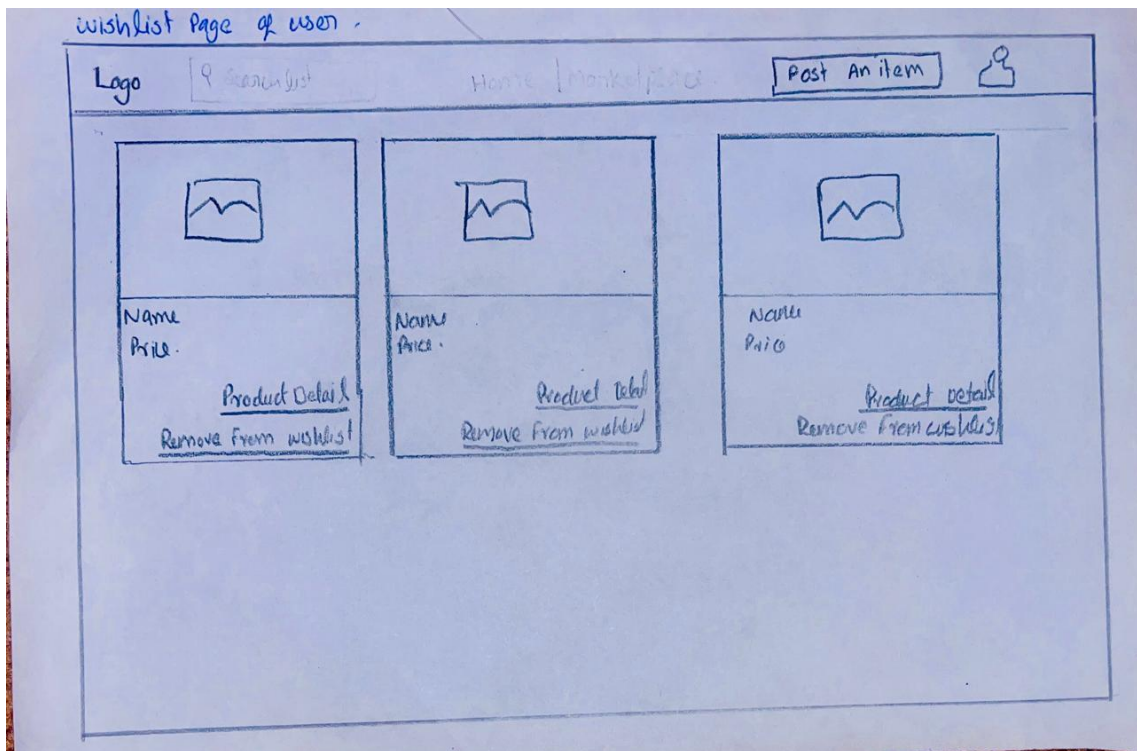
Description:

This user profile page displays personal information and allows users to manage their account details. It provides a space for users to present themselves to the marketplace community with customizable bio information.

Key Features:

- **Update Profile Picture:** Profile image upload and display functionality
- **Update Personal Information:** User name and email information display

4.11 Wishlist viewing page



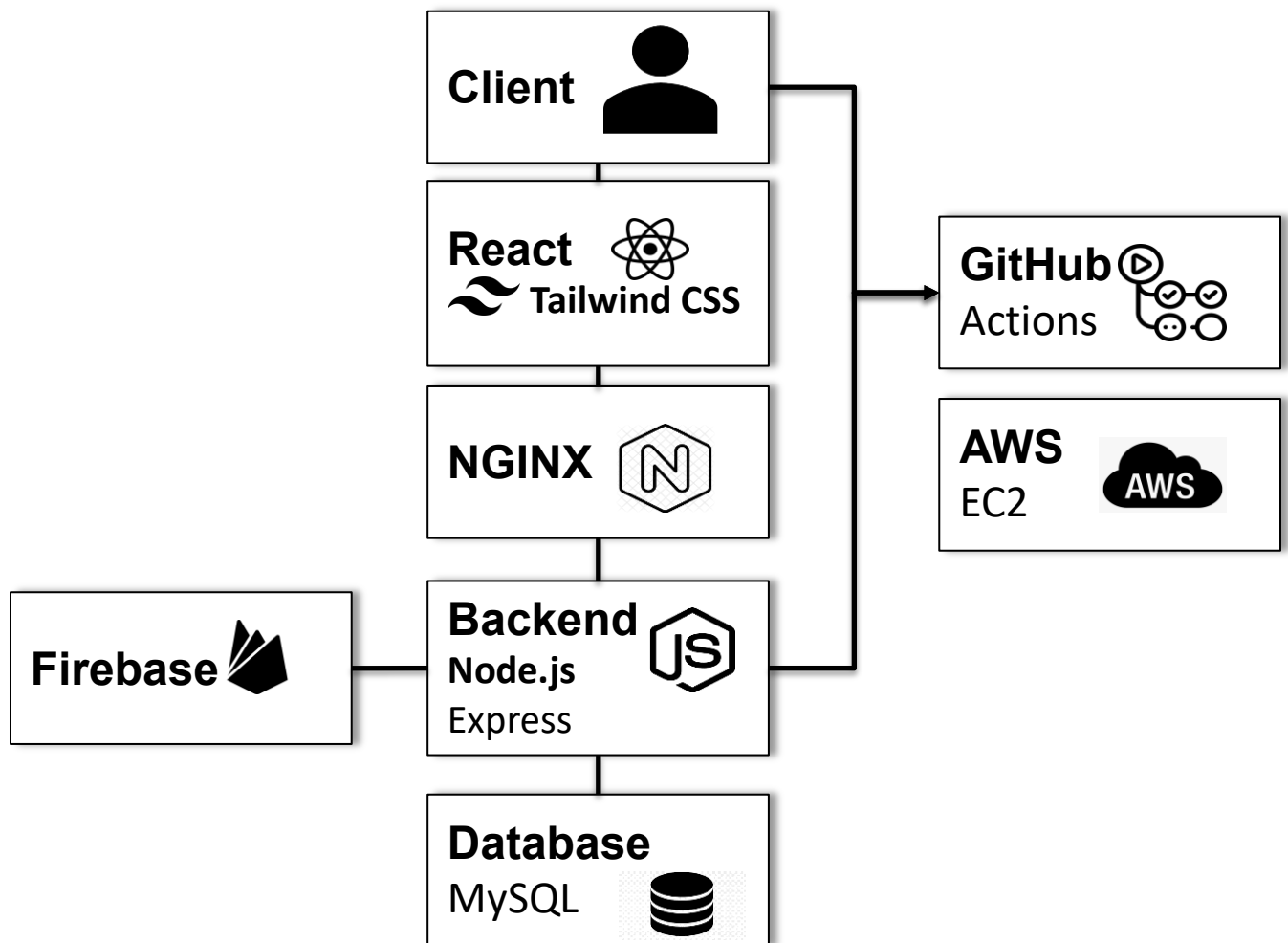
Description:

This wishlist page display all the products wishlisted by the user . This can be used by the users to view their favorite items later.

Key Features:

- **Remove from wishlist:** Users can remove the wishlisted product if its not needed anymore.

5. NEW HIGH LEVEL ARCHITECTURE, DATABASE ORGANIZATION



5.1 DB ORGANIZATION

Our application uses a relational database (MySQL 8.0) deployed on AWS EC2. The main tables are:

- **Users:** Stores registered users (buyers, sellers, moderators)
- **ProductListings:** Contains items for sale, their metadata, and creator
- **MediaFiles:** Holds file paths for uploaded images/videos

- Chat: Links user pairs to specific product listings with message context
- Messages: Stores all in-app communications between users
- Favorites: Tracks which products are saved by users
- Reports: Allows users to report inappropriate listings or users
- Search History: Logs filter parameters used by users

5.2 MEDIA STORAGE

Media (images) will be stored externally using Firebase Cloud Storage, not in the database as BLOBs. Only the image URL will be stored in the database.

5.3 SEARCH/FILTER ARCHITECTURE AND IMPLEMENTATION

In backend, we will use SQL LIKE to search the user input from product title, description, and tags. After that, we will use SQL WHERE to filter the results. The user can filter by category, price range, condition (new or used), and location. These filters will help users to find the product more easily. Sorting will be done on server side using ORDER BY clause, for example by date or price.

5.4 APIS

Listings API

- GET /api/listings – Retrieve a list of all product listings
- POST /api/listings – Create a new product listing
- PUT /api/listings/{id} – Update a listing by ID
- GET /api/listings/{id} – Retrieve a specific product listing
- DELETE /api/listings/{id} – Delete a listing by ID
- GET /api/listings/search – Search for listings by keyword

Users API

- POST /api/users – Register a new user
- GET /api/users – Get all users
- GET /api/users/{id} – Get user by ID
- PUT /api/users/{id} – Update user profile info (name, picture, etc.)

Favourites API

- POST /api/favourites – Add a product to favourites
- GET /api/favourites/{userId} – Get a user's favourites
- DELETE /api/favourites/{productId} – Remove product from favourites

Reports API

- POST /api/reports – File a report
- GET /api/reports/{id} – Get details of a specific report
- GET /api/reports/user/{userId} – Get all reports of a specific user
- GET /api/reports – Retrieve all reports

Chat API

- GET /api/chat/user/{userId} – Get all chats for a user (includes receiver name, product image, last message, etc.)

Messages API

- GET /api/messages/chat/{chatId} – Retrieve all messages for a specific chat
- POST /api/messages – Send a message (stores sender_id, receiver_id, product_id, content)

5.5 NON-TRIVIAL ALGORITHM

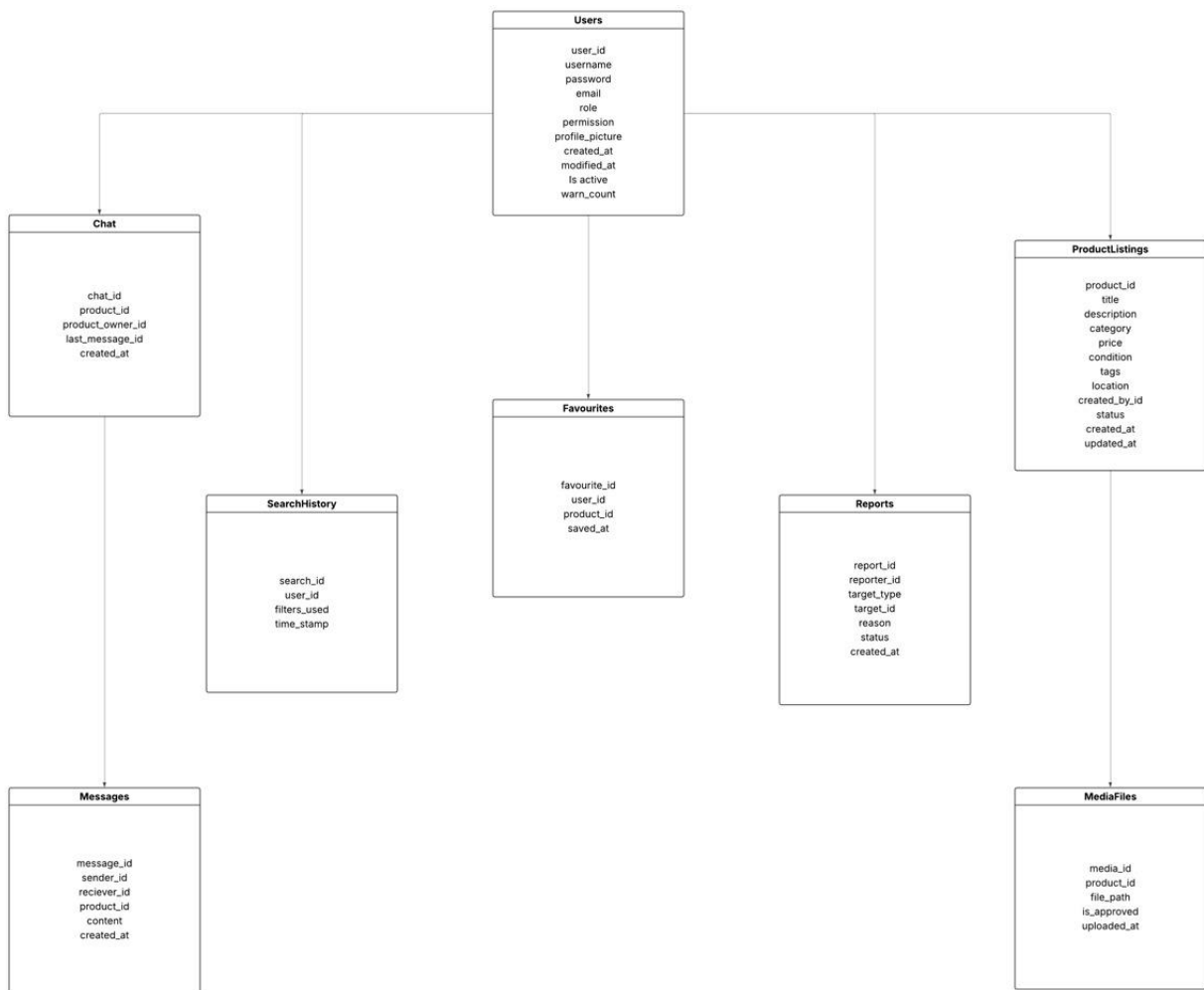
Our application does not implement any non-trivial or custom algorithms. All logic is handled through standard CRUD operations and SQL queries.

5.6 TOOLS AND FRAMEWORKS

We continue using the tools and frameworks listed in Milestone 1. There is no modification in them.

6. HIGH LEVEL UML DIAGRAMS

This section will contain UML diagrams describing the system architecture, class relations, and other modeling views.



7. IDENTIFY ACTUAL KEY RISKS FOR YOUR PROJECT AT THIS TIME

7.1 SKILL RISK

7.1.1 Lack of skill

Risk: We have 2 team members for frontend and 3 for backend and 1 GitHub Master. Few team members don't have any experience in developing real time applications and they are learning new technologies and there is a risk involved that incase those skills of a certain team member are not sufficient to build the project.

Solution: Pair less experienced developers with more knowledgeable team members for mentorship. Allocate initial project time (e.g., first week) for crash-course tutorials, prototyping, and tech exploration.

7.2 SCHEDULE RISKS

7.2.1 Time Slots Not Overlapping

Risk: Team members have different schedules due to other projects, classes, and jobs, which risks disrupting communication and project progress.

Solution: Meetings will be planned in advance to accommodate everyone's schedules and ensure full participation.

7.2.2 Team Member's Ability to Work

Risk: A team member might suddenly be unable to work, for example, due to health issues.

Solution: By planning ahead, work can be reassigned to other team members who can take over the affected workload.

7.2.3 Overcommitment of Team Members

Risk: Due to team members working on multiple projects or studying for exams, there might be weeks when some members can barely contribute to this project, potentially causing delays.

Solution: The team will communicate weekly about each member's available time and distribute tasks accordingly. Lost time will be compensated by increased effort in the following weeks.

7.3 TEAMWORK RISKS

7.3.1 Miscommunication and Conflicts Within the Group

Risk: During the project, conflicts or miscommunication within the group may occur.

Solution: To address such issues, a discussion round will be established where problems can be resolved collaboratively by team members.

7.3.2 Uneven Contribution

Risk: Some members may contribute significantly less, leading to frustration and imbalance in work distribution.

Solution: Set transparent weekly goals for each member. Log contributions through version control and task tracking tools.

7.4 Technical Risks

7.4.1 Exceeding AWS Student Credits

Risk : The project is being developed using AWS student credits These credits have a limited balance, and if the usage exceeds this limit, the team may incur unexpected charges or be forced to halt cloud-based development.

Solution: Regularly monitor credit usage through the AWS Billing & Cost Management Dashboard. Keep EC2 instances stopped when not actively being used and delete unused volumes or snapshots to reduce storage costs.

8. PROJECT MANAGEMENT

Our workflow for M2 Document tasks was as follows:

- On Friday after meeting, we discuss what needs to be done.
- Each member gets assigned tasks (assigned in Trello to track) and works on them until the team meeting on Wednesday.
- On Wednesday, we check the work done together and discuss what to change until the Friday meeting.

The workflow for the M2 Vertical SW Prototype and future tasks will be:

- At the end of the Friday meeting, discuss and assign tasks.
- The team lead updates Trello accordingly, so team members can check what work they were assigned.
- On Wednesday the team meets, reviews the completed work, discusses it, and decides on changes before Friday.
- If any problem occurs, the team will meet online or offline to find a solution.
- For communication, the team uses WhatsApp and Google Meet.