Swe573 | Nesibe Betül Döner | 2023719039

Eureka Nexus

eurekanexus.onrender.com

github.com/betulnesibe/swe573-software-development-practice

github.com/betulnesibe/swe573-software-development-practice/releases/tag/v0.9

21.12.2024

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 Fall 2024 semester project reported in this report, I Nesibe Betül Döner declare that:

- I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the Fall 2024 semester.
- All the material that I am submitting related to my project (including but not limited to the project repository, the final project report, and supplementary documents) have been exclusively prepared by myself.
- I have prepared this material individually without the assistance of anyone else with the exception of permitted peer assistance which I have explicitly disclosed in this report.

Nesibe Betül Döner

N Betul

1. TABLE OF CONTENTS

2. REQUIRED INFORMATION TO ACCESS AND TEST THE DEPLOYED SYSTEM

2.1. Site URL

https://eurekanexus.onrender.com/

2.2. Users Table

Created users for the system are as follows:

Username	Name	Surname	Email	Password
ironman	Tony	Stark	tony@stark.com	Avengers123.
captainamerica	Steve	Rogers	steve@america.com	Avengers123.
loki	Loki	Odinson	loki@godofmischief.com	Mischief123.
thor	Thor	Odinson	thor@godofthunder.com	Thunder123.
admin			admin@admin.com	admin123

2.3. Admin Panel Access

Access to the admin panel can be achieved by following these steps:

1) The credentials admin as the username and admin123 as the password should be used to log in.
2) Upon logging in, the page will automatically be redirected to the home page.
3) The URL must then be manually changed to https://eurekanexus.onrender.com/admin/ to access the admin panel.

3. DECLARATION

It is declared that all the work in this project was done in compliance with the honour code stated on the title page of this report. The project deliverables, such as the codebase, documentation, and report content, were created independently without unauthorized help.

During this project, some third-party resources to is used to guide and solve problems:

1) *ChatGPT (OpenAI)*: Used to get ideas, explanations, and help with code generation, and debugging.
2) *Django documentation*: Used as a guide for using Django's built-in features.
3) *Stack Overflow*: Checked for solutions to coding issues and best practices.
4) *Reddit*: Checked for solutions to coding issues and best practices.

All third-party tools, code, and libraries were used according to their licenses.

4. PROJECT DETAILS

4.1. Overview

Eureka Nexus is a web platform designed to identify unknown human-made objects through a collaborative, community-driven approach. The name of the project is selected to reflect its purpose. While "Eureka" signifies discovery, "Nexus" represents connection and collaboration.

Eureka Nexus creates an efficient platform in order to identify objects where users can:

- Post images of unidentified objects and provide detailed descriptions using attributes.
- Collaborate with other users through threaded comments and voting.
- Utilize semantic tags fetched from Wikidata for enhanced categorization and searchability.

The technological stack of the Eureka Nexus is as follows:

- Frontend and Backend: Python with Django framework
- Database: PostgreSQL
- Containerization: Docker
- Deployment: Render.com
- Version Control: Git and Github

4.2. Software Requirements Specification

4.2.1. Introduction

4.2.1.1. Purpose:

The purpose of this project is to develop Eureka Nexus, a web-based platform where users can post unknown human-made objects and seek community assistance in identifying them through interaction. By detailed object creation and fostered collaboration, the platform enables detailed discussions, multimedia uploads, semantic tagging, and community-driven identification of objects.

4.2.1.2. Scope:

The platform includes features for user registration and profile management, object creation, community engagement (commenting and voting), moderation, search (basic and advanced), and a reputation system. The system integrates with external APIs (Wikidata and location), and it is built for scalability and usability. The system emphasizes community-driven identification through threaded discussions, with features like anonymous posting, semantic tagging, and reputation incentives.

4.2.2. Functional Requirements

This section describes the functionalities that the system will support.

4.2.2.1. Account Registration Management

4.2.2.1.1. User accounts must be created with a name, surname, date of birth, email address, and password.

4.2.2.1.2. Passwords must be at least 8 characters long and include numbers, letters, and special characters.

4.2.2.1.3. Password verification must be enforced during account creation.

4.2.2.1.4. After creating an account, users should be directed to an Edit Profile page where they can optionally add a profile picture, bio, and tags. If users choose not to add anything:

4.2.2.1.4.1. A default profile picture will be assigned.

4.2.2.1.4.2. The bio will default to: "Hello, I am a member of this platform."

4.2.2.1.4.3. No tags will be assigned unless the user explicitly adds them.

4.2.2.1.5. Password recovery options must be provided.

4.2.2.2. User Management

4.2.2.2.1. Users must be able to log in and log out securely using their email address and password.

4.2.2.2.2. Users must be able to delete their accounts.

4.2.2.2.3. A Profile screen should be provided where it displays:

4.2.2.2.3.1. Posts, comments, and upvotes/downvotes.

4.2.2.2.3.2. Profile picture, bio, tags, badges, and an edit profile button.

4.2.2.2.4. Users must be able to edit their profile information, including passwords, emails, profile pictures, bios, and tags, through an Edit Profile screen accessible via their profile.

4.2.2.2.5. Users must be able to view other users' profile screens displaying their posts and comments.

4.2.2.3. Posting System

4.2.2.3.1. Posts must include:

4.2.2.3.1.1. Title: Text field.

4.2.2.3.1.2. Description: Text field.

4.2.2.3.1.3. Image: Upload field supporting gif, jpeg, jpg, png, svg, webp, and heic formats.

4.2.2.3.1.4. Attributes: A dynamic list allowing users to select 0, 1, many, or all attributes. Selected attributes reveal input fields as following:

4.2.2.3.1.4.1. Condition, Shape, Colour, Material, Texture, Smell, Taste, Pattern, Functionality, Time Period, Object Domain, Hardness, Elasticity, and Transparency: Dropdown with "Other" option enabling manual input.

4.2.2.3.1.4.2. Origin, Value, Origin of Acquisition, Image Description, Icon Description, Print Description, Markings, and Brand: Text fields.

4.2.2.3.1.4.3. Size: Text fields for height, width, depth and a dropdown for approximate/exact and metrics.

4.2.2.3.1.4.4. Weight: Text field and a dropdown for approximate/exact.

4.2.2.3.1.4.5. Location: Text field that searches for locations.

4.2.2.3.1.5. Other Multimedia: Upload field.

4.2.2.3.1.6. Semantic Tags: Text field that searches on Wikidata to add tags.

4.2.2.3.1.7. Status: Defaulted to "Unknown" (editable post-creation).

4.2.2.3.1.8. Post owner: Anonymous or the user.

4.2.2.3.2. Posts must be viewable by all users, including non-registered users.

4.2.2.3.3. Post owners can update the status from "Unknown" to "Solved."

4.2.2.3.4. Users cannot edit their posts but can delete them.

4.2.2.3.5. Deleted user accounts leave posts in the database as "anonymous."

4.2.2.3.6. Posts include upvote and downvote counters, initialized at zero.

4.2.2.4. Commenting and Interaction

4.2.2.4.1. Users can comment on posts.

4.2.2.4.2. Comments can be tagged as "question" or "hint" (optional). Tagged comments display markers.

4.2.2.4.3. Post owners can tag a comment as “answer”.

4.2.2.4.4. Threaded discussions allow users to respond to comments.

4.2.2.4.5. Users can upvote/downvote posts and comments.

4.2.2.4.6. Comments cannot be edited but can be deleted. Deleted comments display as "[Deleted]."

4.2.2.4.7. Posts and comments can be reported. Reporting opens a modal dialog where users must specify the reason (required) before submitting.

4.2.2.5. Moderation

4.2.2.5.1. Moderators ensure content adheres to platform policies.

4.2.2.5.2. Moderators review grouped reports (e.g., all reports for a post are grouped).

4.2.2.5.3. Actions include issuing warnings after up to three valid reports or removing content after three valid reports.

4.2.2.5.4. Moderators can update post statuses to "Solved" or "Unknown."

4.2.2.6. Search and Query

4.2.2.6.1. Basic search filters posts by keywords in titles.

4.2.2.6.2. Advanced search filters posts by attributes or semantic tags.

4.2.2.6.3. Search results can be sorted by date or alphabetically.

4.2.2.7. Reputation System

- Users earn badges based on activity (posting, commenting, upvotes). Example badges: "First/Fifth Post”, "First/Fifth Comment", "First/Tenth Upvote”.

4.2.2.8. Feed and Notifications

4.2.2.8.1. Users can follow other users and semantic tags for a personalized feed.

4.2.2.8.2. Feed displays posts from followed users or tags.

4.2.2.8.3. Notifications alert users to comments, votes, and new posts.

4.2.2.8.4. Notification preferences can be customized (e.g., email, in-app notifications, or on posts, replies, or votes).

4.2.2.9. Data Management and Integration

4.2.2.9.1. Integration with Wikidata for semantic tagging.

4.2.2.9.2. Attributes (e.g., dates, weights) must be stored in standardized formats for computational processing.

4.2.3. Non-Functional Requirements

4.2.3.1. Performance

The system must handle concurrent users, multimedia uploads, and real-time interactions without significant latency.

4.2.3.2. Scalability

The system must support growth in users, posts, and multimedia.

4.2.3.3. Security

The system must encrypt sensitive data and prevent unauthorized access.

4.2.3.4. Accessibility

The system should acknowledge partial accessibility for visually impaired users.

4.2.3.5. Usability

The system should ensure intuitive navigation for all functionalities.

4.2.3.6. Data Integrity

The system should maintain reliable data storage for profiles, posts, and interactions.

4.2.3.7. Compliance

The system should adhere to regulations like GDPR.

4.2.4. Constraints and Assumptions

4.2.4.1. Constraints

4.2.4.1.1. The system relies on an external API (for image search) and Wikidata integration, which might affect system availability if these services are down.

4.2.4.2. Assumptions

4.2.4.2.1. Moderators and original posters are responsible for the accuracy and credibility of the information provided.

4.2.4.2.2. System limitations include lack of full accessibility for visually impaired users.

4.2.4.2.3. Posts remain publicly accessible even after account deletion, ensuring the continuity of the knowledge base.

4.2.4.2.4. 3.2.4. There is no multilingual support and descriptions will be limited to specific languages that the post owner uses

4.2.5. System Architecture

4.2.5.1. Architecture

4.2.5.1.1. Frontend and Backend: Python with Django framework

4.2.5.1.2. Database: PostgreSQL

4.2.5.1.3. Containerization: Docker

4.2.5.1.4. Deployment: Render.com

4.2.5.1.5. Version Control: Git and Github

4.2.5.2. Integration

4.2.5.2.1. External APIs for location search and Wikidata.

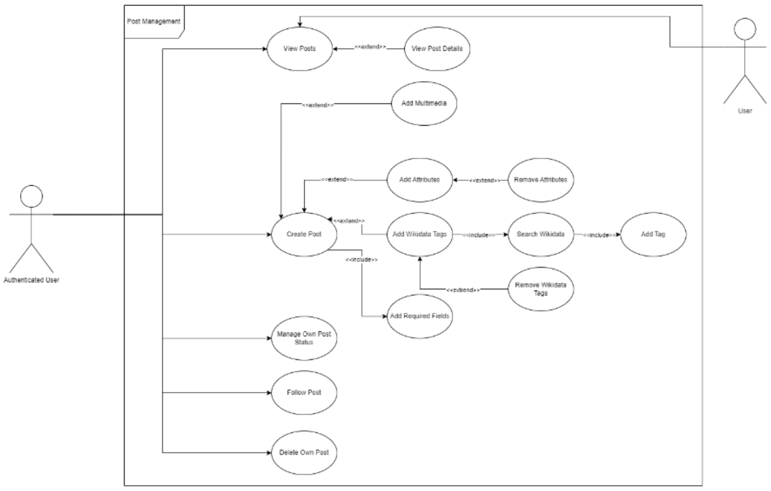
4.2.5.3. Data Processing

4.2.5.3.1. Computational handling for advanced filtering.

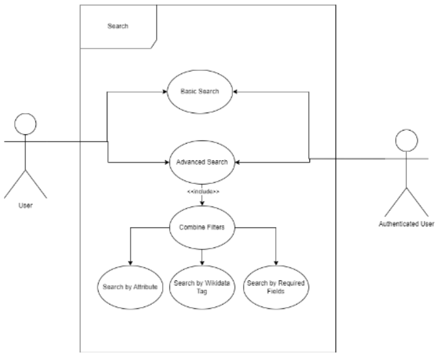
4.3. Design Documents

4.3.1. Use Case Diagrams

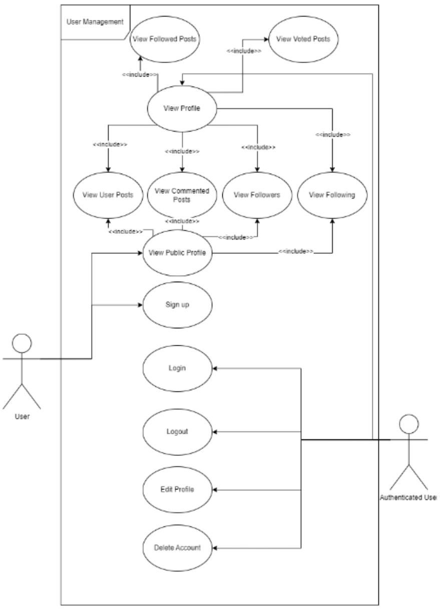
4.3.1.1. Post Management



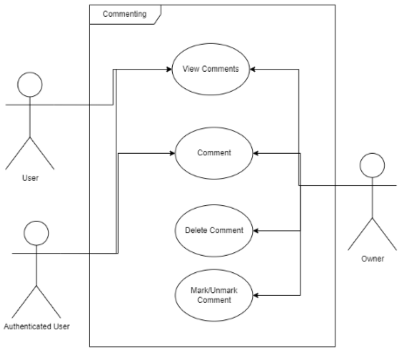
4.3.1.2. Search



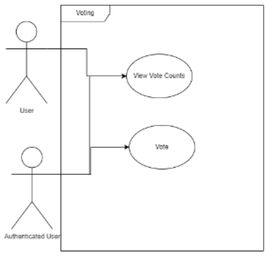
4.3.1.3. User Management



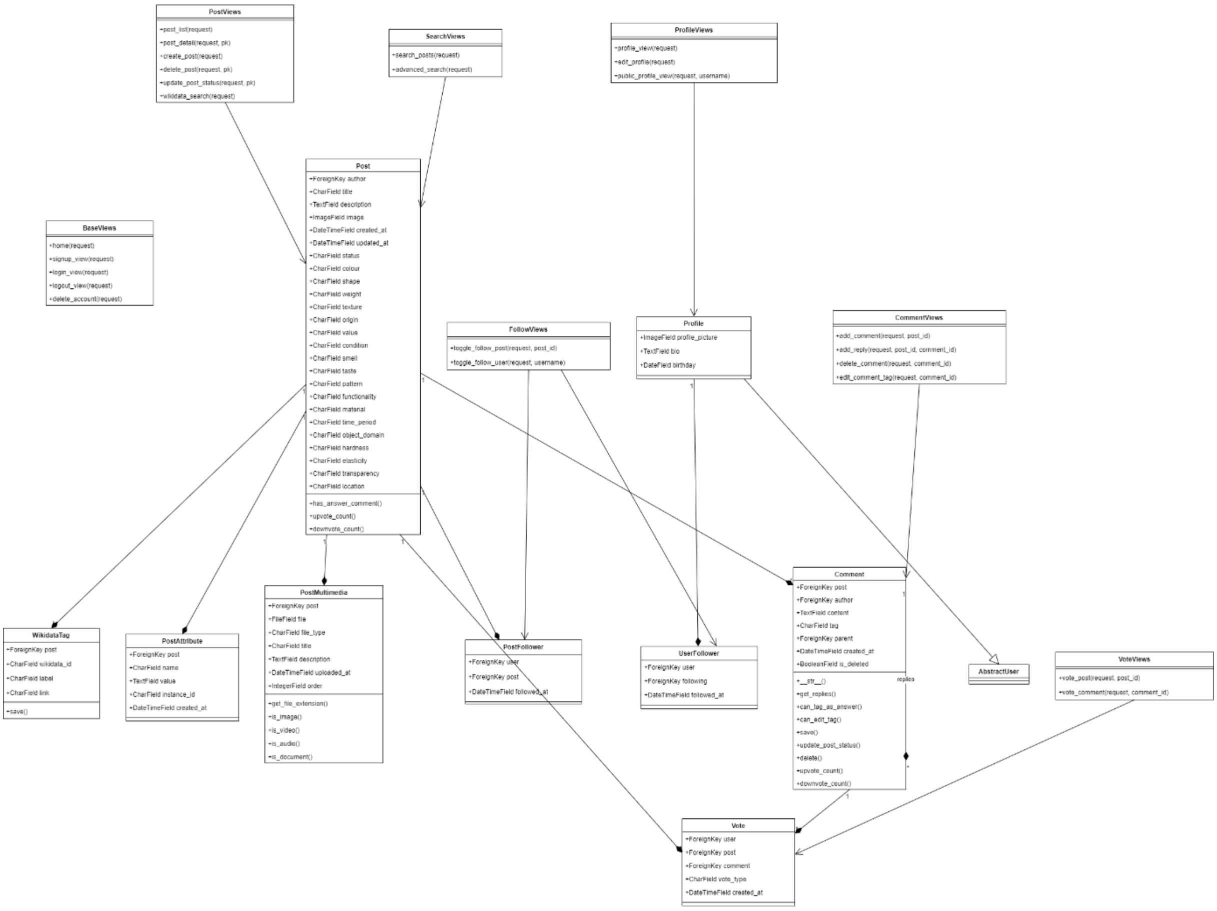
4.3.1.4. Commenting



4.3.1.5. Voting

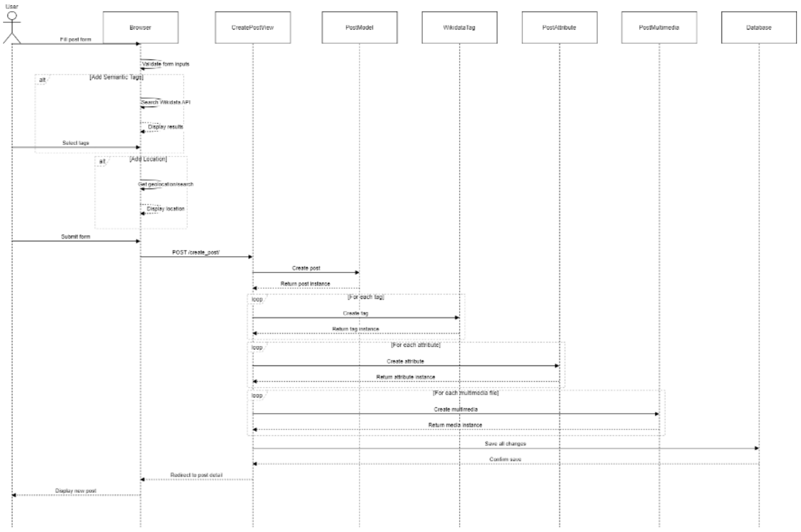


4.3.2. Class Diagram

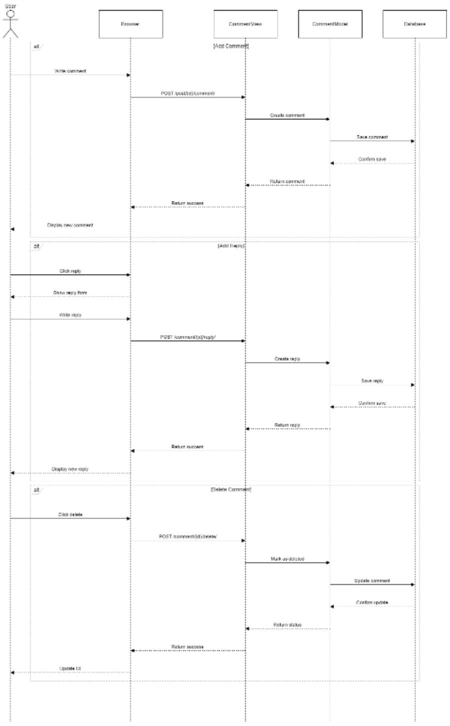


4.3.3. Sequence Diagrams

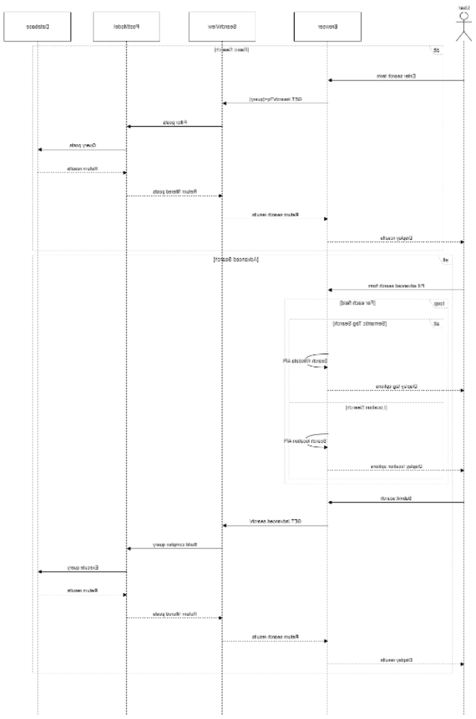
4.3.3.1. Create Post



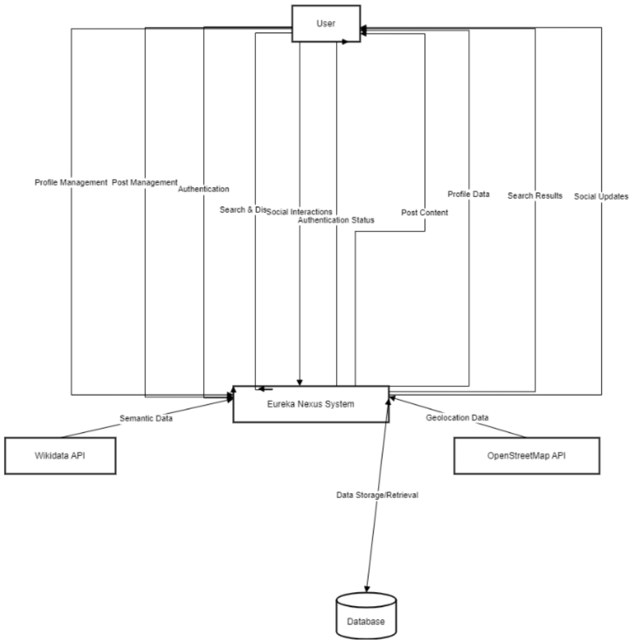
4.3.3.2. Commenting



4.3.3.3. Searching

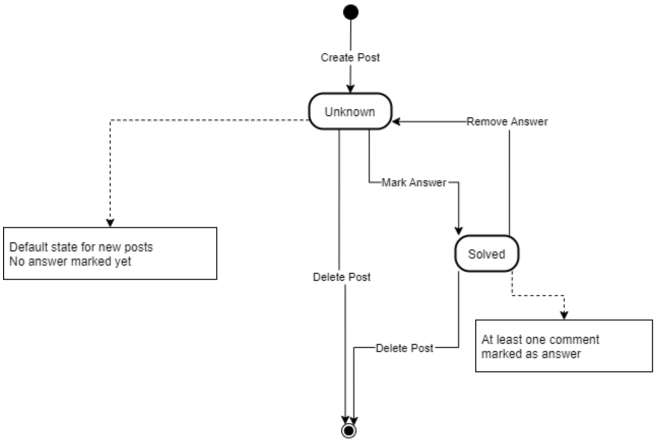


4.3.4. Data Flow Diagram

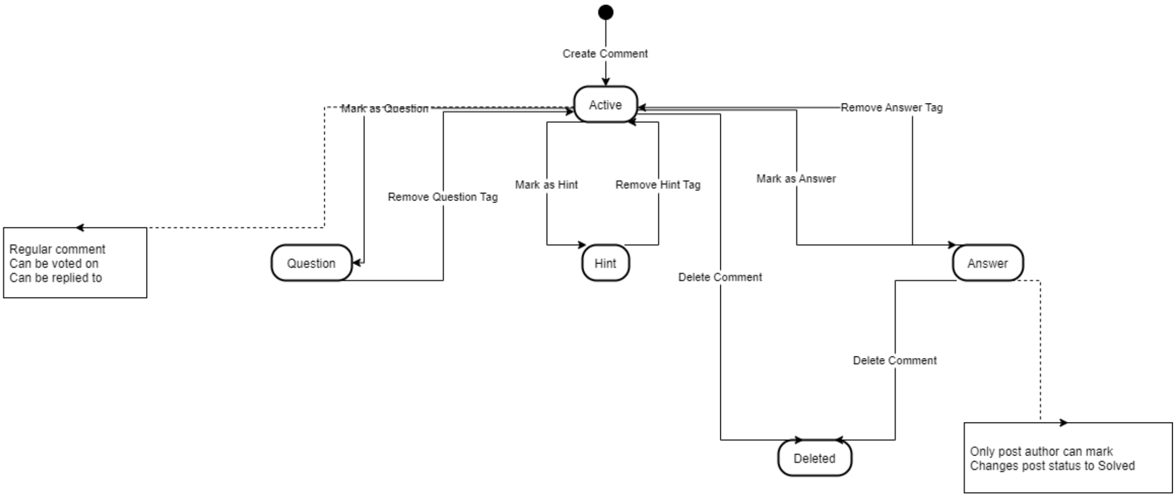


4.3.5. State Transition Diagrams

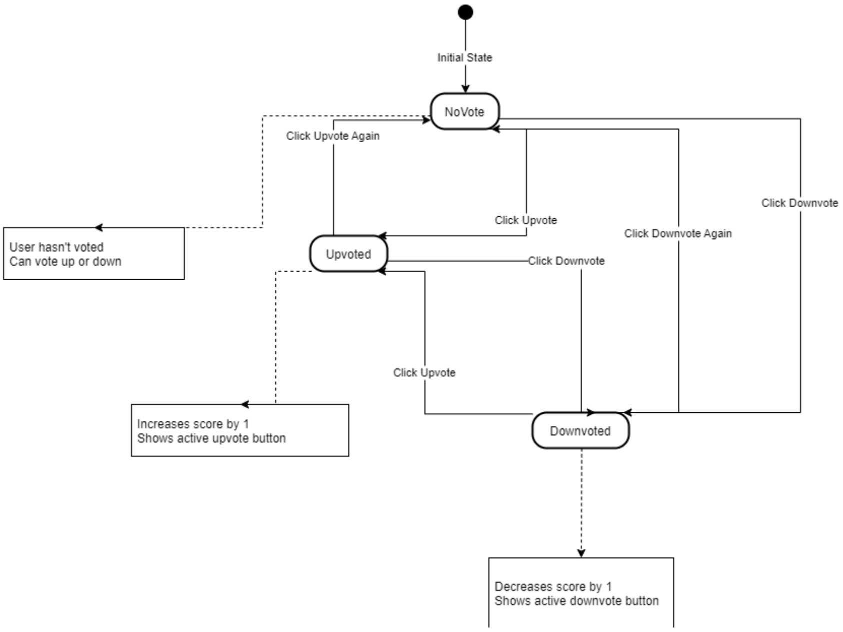
4.3.5.1. Post Status



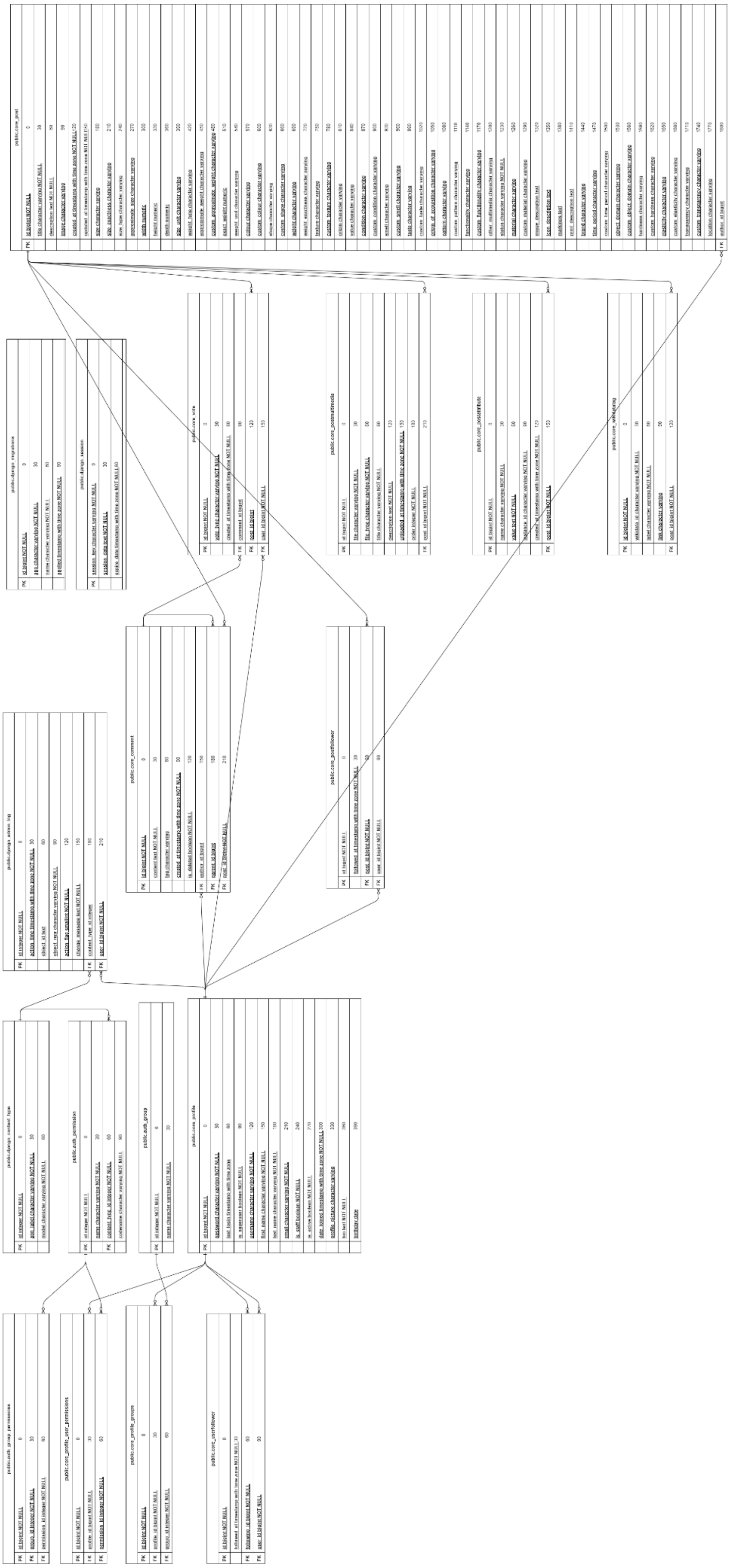
4.3.5.2. Comment



4.3.5.3. Vote


Initial State
NoVote
Click Upvote Again
Click Upvote
Click Downvote Again
Click Downvote
User hasn't voted
Can vote up or down
Upvoted
Click Downvote
Click Upvote
Increases score by 1
Shows active upvote button
Downvoted
Decreases score by 1
Shows active downvote button

4.3.6. Entity Relationship Diagram (taken from PostgreSQL)



4.4. Project Status

	Requirement	Status	Documented	Tested	Deployed	Notes
Account Registration Management	4.2.2.1.1.	Completed	Completed	Completed	Completed	Date of birth is not added deliberatively.
	4.2.2.1.2.	Completed	Completed	Completed	Completed	
	4.2.2.1.3.	Completed	Completed	Completed	Completed	
	4.2.2.1.4.	Completed	Completed	Completed	Completed	Tags are not added.
	4.2.2.1.5.	Completed	Completed	Completed	Completed	
User Management	4.2.2.2.1.	Completed	Completed	Completed	Completed	
	4.2.2.2.2.	Completed	Completed	Completed	Completed	
	4.2.2.2.3.	Completed	Completed	Completed	Completed	Tags are not added.
	4.2.2.2.4.	Completed	Completed	Completed	Completed	Tags are not added.
	4.2.2.2.5.	Completed	Completed	Completed	Completed	
Posting System	4.2.2.3.1.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.3.2.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.3.3.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.3.4.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.3.5.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.3.6.	Partially completed	Completed	Not Completed	Completed	
Commenting and Interaction	4.2.2.4.1.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.4.2.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.4.3.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.4.4.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.4.5.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.4.6.	Not completed	Not completed	Not completed	Not completed	
Moderation	4.2.2.5.1.	Not completed	Not completed	Not completed	Not completed	
	4.2.2.5.2.	Not completed	Not completed	Not completed	Not completed	
	4.2.2.5.3.	Not completed	Not completed	Not completed	Not completed	
	4.2.2.5.4.	Partially completed	Completed	Not Completed	Completed	
Search and Query	4.2.2.6.1.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.6.2.	Partially completed	Completed	Not Completed	Completed	It has some bugs.
	4.2.2.6.3.	Not completed	Not completed	Not completed	Not completed	

Reputation System	4.2.2.7.1.	Not completed	Not completed	Not completed	Not completed	
Feed and Notifications	4.2.2.8.1.	Partially completed	Completed	Not Completed	Completed	Personalized feed is not implemented.
	4.2.2.8.2.	Not completed	Not completed	Not completed	Not completed	
	4.2.2.8.3.	Not completed	Not completed	Not completed	Not completed	
	4.2.2.8.4.	Not completed	Not completed	Not completed	Not completed	
Data Management and Integration	4.2.2.9.1.	Partially completed	Completed	Not Completed	Completed	
	4.2.2.9.2.	Partially completed	Completed	Not Completed	Completed	

* If a requirement is documented and deployed but not tested, it means it is manually tested but unit tests are not created.

4.5. Deployment Status

The project has two branches regarding deployment.

4.5.1. Eureka-nexus-m Branch

- Deployment platform: Google Cloud Run
- Deployment URL: https://eureka-nexus-391682758262.us-central1.run.app/
- Branch URL: https://github.com/betulnesibe/swe573-software-development-practice/tree/eureka-nexus-m
- Explanation: This branch contains the dockerized version of the project. While dockerization was successfully implemented, the deployment on Google Cloud Run encountered issues related to multimedia management. These bugs hindered the seamless functioning of the project in this environment. To ensure a functional demo and deployment, rapid adjustments were necessary, as the absence of a working deployment is unacceptable from a customer's perspective. Consequently, the project was migrated to render.com for deployment.

4.5.2. Main Branch

- Deployment platform: render.com
- Deployment URL: https://eurekanexus.onrender.com/
- Branch URL: https://github.com/betulnesibe/swe573-software-development-practice
- Explanation: The main branch of the project is deployed on render.com, which is the demoed version of the project. Since render.com does not require Docker, dockerization was not utilized for this deployment. Instead, the platform-specific deployment instructions were followed to ensure a smooth and functional release.

4.6. Installation Instructions

4.6.1. Installation of the Main Branch (Deployed Version)

Since the main branch does not dockerized version of the project due to the issue mentioned above, the project installation of the deployed version can be conducted as follows (Note: Terminal commands may vary for MacOS/Linux system, use equivalents for these systems):

4.6.1.1. Prerequisites

- Python
- Pip
- PostgreSQL
- Git

4.6.1.2. Clone the Repository

Copy and paste the following on terminal:

```
git clone https://github.com/betulnesibe/swe573-software-development-practice.git
cd swe573-software-development-practice
```

4.6.1.3. Create a Virtual Environment

Copy and paste the following on terminal:

```
python -m venv venv
\venv\Scripts\activate
```

4.6.1.4. Install Dependencies

Copy and paste the following on terminal:

```
pip install -r requirements.txt
```

4.6.1.5. Create PostgreSQL Database

- Using the pgAdmin, create a blank database named as “eureka_nexus_db”.
- Using the PostgreSQL command prompt enter the following:
  ```
  CREATE DATABASE eureka_nexus_db
  ```

Make sure PostgreSQL service is running before starting the application

4.6.1.6. Setup Environment Variables

1) Create a “.env” file in the root directory and add the following configuration to it using text editor or notepad.

```
## Database Configuration
DB_USER= ## add username of the PostgreSQL database
DB_PASSWORD= ## add password of the PostgreSQL database
## Django Configuration
SECRET_KEY= ## add secret key for the project
DEBUG=True
```

2) In settings.py file in the project directory, change the following:

- ALLOWED_HOSTS= [127.0.0.1, 127.0.0.1:8000, localhost]
- Uncomment the first “DATABASES” dictionary under the # Database comment.
- Comment out the second “DATABASES” dictionary.

** Note that if the deployment will be done on render.com, steps for environment variables should be followed from this link (https://render.com/docs/deploy-django#updating-an-existing-django-project).

4.6.1.7. Run Migrations

Copy and paste the following on terminal:

```
python manage.py makemigrations
python manage.py migrate
```

4.6.1.8. Create a Superuser (Admin)

Copy and paste the following on terminal:

```
python manage.py createsuperuser
```

4.6.1.9. Run Development Server

Copy and paste the following on terminal:

```
python manage.py runserver
```

After following these steps, the application should now be running from http://127.0.0.1:8000/. The admin page can be accessed from http://127.0.0.1:8000/admin .

4.6.2. Installation of the Eureka-nexus-m Branch

4.6.2.1. Prerequisites

- Docker
- Docker Compose
- Git

4.6.2.2. Clone the Repository and Switch Branch

Copy and paste the following on terminal:

```
git clone https://github.com/betulnesibe/swe573-software-development-practice.git
cd swe573-software-development-practice
git checkout eureka-nexus-m
```

4.6.2.3. Setup the Environment Variables

Follow step 4.6.1.6.

4.6.2.4. Build and Run with Docker Compose

Copy and paste the following on terminal:

```
docker-compose up --build
```

4.6.2.5. Run Initial Migrations

Copy and paste the following on terminal:

```
docker-compose exec web python manage.py migrate
```

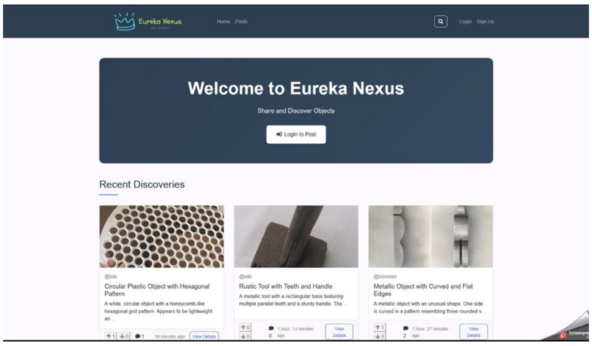
4.6.2.6. Create a Superuser (Admin)

Copy and paste the following on terminal:

```
docker-compose exec web python manage.py createsuperuser
```

4.7. User manual

4.7.1. Guest Home Page



Welcome to Eureka Nexus

Share and Discover Objects

Recent Discoveries

@loki
Circular Plastic Object with Hexagonal Pattern
A white, circular object with a honeycomb-like hexagonal grid pattern. Appears to be lightweight an...
↑1 ↓0 1 39 minutes ago

@loki
Rustic Tool with Teeth and Handle
A metallic tool with a rectangular base featuring multiple parallel teeth and a sturdy handle. The ...
↑0 ↓0 0 1 hour, 14 minutes ago

@ironman
Metallic Object with Curved and Flat Edges
A metallic object with an unusual shape. One side is curved in a pattern resembling three rounded s...
↑1 ↓0 2 1 hour, 27 minutes ago

When visiting eurekanexus.onrender.com, the Home Page welcomes visitors with an overview of the platform's purpose and recent discoveries.

4.7.1.1. Purpose

The Home Page serves as the entry point for all users, providing an overview of recent discoveries and a way to explore or contribute to the platform.

4.7.1.2. Key Features

4.7.1.2.1. Welcome Banner:

- Displays a welcoming message: "Welcome to Eureka Nexus."
- Encourages users to "Share and Discover Objects."
- Includes a "Login to Post" button for quick access to the login page.

4.7.1.2.2. Navigation Bar:

- Located at the top of the page, offering links to:
  - Home: Redirects users to this main page.
  - Posts: Allows browsing or creating posts.
  - Search Bar: Enables users to search for specific objects.
  - Login/Signup: Links for user authentication.

4.7.1.2.3. Recent Discoveries Section:

- Displays a list of the latest posts made by users.
- Each card includes:
  - Image: A preview of the object.
  - Title and Description: A brief description of the object.
  - Posted By: The username of the contributor redirecting to the contributor’s profile.
  - Metadata: Provides additional details on voting, comments.

4.7.1.3. How to Use

1) Explore Recent Discoveries: Scroll through the recent posts. Click on any post to view detailed information and interact.
2) Search for Content: Use the search bar in the navigation bar to find specific items by title or use the advanced search button on the search field to conduct advanced search.
3) Login to Post: Use the "Login to Post" button to access the login page.

4.7.1.4. Features for Guests

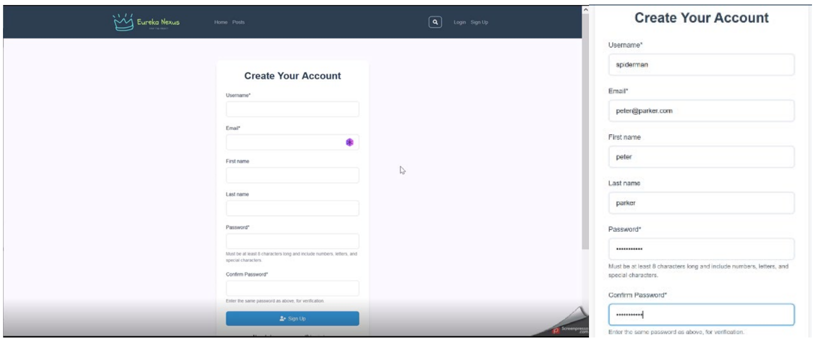
4.7.1.4.1. A guest can:

- View Posts: Browse through posts shared by other users.
- Access Comments and Voting Counts: View comments and the total votes on posts.
- Perform Searches: Use both basic and advanced search functionalities.

4.7.1.4.2. Guests cannot:

- Create posts.
- Interact with posts (e.g., by commenting or voting).
- Follow users or posts.

4.7.2. Sign Up Page



Create Your Account

Username*

Email*

First name

Last name

Password*

Must be at least 8 characters long and include numbers, letters, and special characters.

Confirm Password*

Enter the same password as above, for verification.

Create Your Account

Username*: spiderman

Email*: peter@parker.com

First name: peter

Last name: parker

Password*: ••••••••••••

Must be at least 8 characters long and include numbers, letters, and special characters.

Confirm Password*: •••••••••••

Enter the same password as above, for verification.

4.7.2.1. Purpose

The Sign-Up Page allows new users to create an account on the Eureka Nexus platform. This page is essential for accessing all platform features, such as creating posts, commenting, voting, and following users or posts.

4.7.2.2. Fields and Validation Rules

4.7.2.2.1. Required Fields:

- Username: Must be unique. If the username is already taken, an error message will prompt: "This username is already taken. Please choose a different one.".
- Email: Must be unique and a valid email address. If the email is already taken, an error message will prompt: "This email address is already in use."
- Password: Must be at least 8 characters long and include a combination of numbers, letters, and special characters.
- Confirm Password: Must match the password entered in the "Password" field. If the passwords do not match, an error message will prompt: "Passwords do not match."

4.7.2.2.2. Optional Fields:

- First Name and Last Name.

4.7.2.3. How to Create an Account

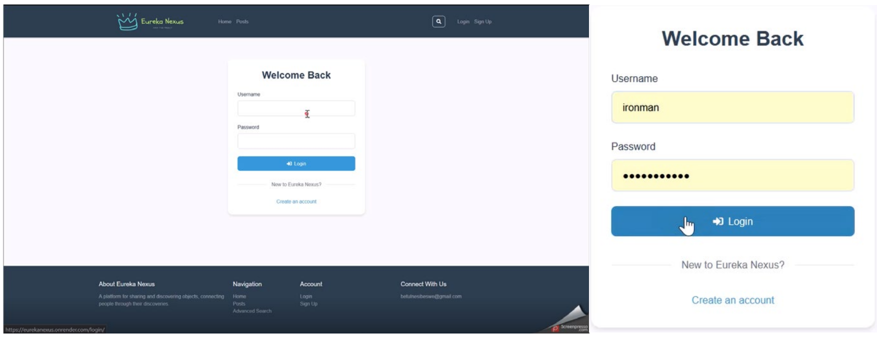
1) Enter your Username and ensure it is unique.
2) Provide a valid Email address.
3) Optionally, fill in your First Name and Last Name.
4) Create a secure Password that meets the criteria.
5) Re-enter the password in the Confirm Password field. Ensure both passwords match.
6) Click the Sign Up button to create your account.
7) Redirect to the Profile Page.

4.7.2.4. Additional Options

4.7.2.4.1. Log in: If there is an account, click the login link to redirect to Login Page.

4.7.2.4.2. Navigation Bar: Use the navigation bar to return to the Home Page or explore posts as a guest.

4.7.3. Login Page



A platform for sharing and discovering objects, connecting people through their discoveries.

Welcome Back

Username: ironman

Password: •••••••••••

4.7.3.1. Purpose

The Login Page allows registered users to access their accounts on the Eureka Nexus platform. By logging in, users can interact with posts, comment, vote, follow other users or posts, and create new content.

4.7.3.2. Fields and Validation Rules

- Username: Enter the unique username associated with the account. It cannot be left blank.
- Password: Enter the password created during account registration. It must match the password associated with the provided username and cannot be left blank.
- If the above credentials are incorrect or missing, the following error message will appear: "Invalid credentials."

4.7.3.3. How to Log In

1) Enter Username in the designated field.
2) Enter Password in the designated field.
3) Click the Login button to access your account.

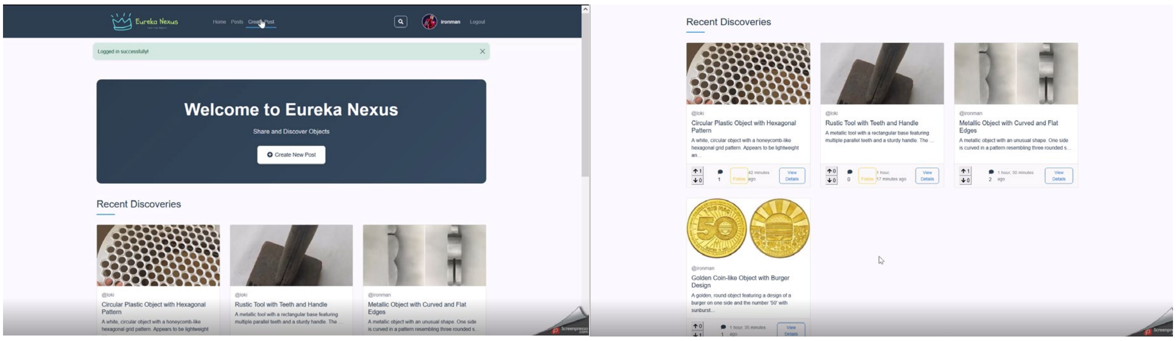
4.7.3.4. Additional Options

4.7.3.4.1. Create an Account: If there is no account, click the Create an account link to redirect to Sign Up Page.

4.7.3.4.2. Navigation Bar: Use the navigation bar to return to the Home Page or explore posts as a guest.

4.7.4. User Home Page

The Home Page of a logged in user redirected from Login Page.



Logged in successfully!

Welcome to Eureka Nexus

Share and Discover Objects

Recent Discoveries

@loki
Circular Plastic Object with Hexagonal Pattern
A white, circular object with a honeycomb-like hexagonal grid pattern. Appears to be lightweight an...
↑1 ↓0 1 42 minutes ago

@loki
Rustic Tool with Teeth and Handle
A metallic tool with a rectangular base featuring multiple parallel teeth and a sturdy handle. The ...
↑0 ↓0 0 1 hour, 17 minutes ago

@ironman
Metallic Object with Curved and Flat Edges
A metallic object with an unusual shape. One side is curved in a pattern resembling three rounded s...
↑1 ↓0 2 1 hour, 30 minutes ago

@ironman
Golden Coin-like Object with Burger Design
A golden, round object featuring a design of a burger on one side and the number '50' with sunburst...
↑0 ↓1 1 1 hour, 35 minutes ago

4.7.4.1. Purpose

The logged-in Home Page serves as the primary dashboard for registered users, offering full platform functionality such as creating posts, interacting with content, and accessing user-specific features.

4.7.4.2. Key Features

4.7.4.2.1. Welcome Banner:

- Displays the welcoming message.
- Includes the Create New Post button for quick access to the post creation page.

4.7.4.2.2. Success Notification:

- A green success message confirms a successful login: "Logged in successfully!"

4.7.4.2.3. Navigation Bar:

- Located at the top of the page, offering links to:
    - Home: Redirects users to this main page.
    - Posts: Allows browsing all posts.
    - Create Post: Directly opens the post creation form.
    - Search Bar: Enables users to search for specific objects.
    - Profile Icon: Displays the logged-in user's profile picture and username.
    - Logout Button: Logs the user out of their account.

4.7.4.2.4. Recent Discoveries Section:

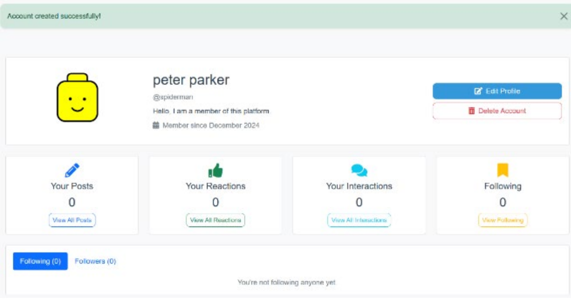
- Displays a list of the latest posts made by users.
- Each card includes:
    - Image: A preview of the object.
    - Title and Description: A brief description of the object.
    - Posted By: The username of the contributor redirecting to the contributor’s profile.
    - Metadata: Provides additional details on voting, comments.
    - Follow Button: Indicates if the post is followed or not.

4.7.4.3. How to Use

1) Create a Post: Click the Create New Post button in the banner or the "Create Post" link in the navigation bar.
2) Explore Recent Discoveries: Scroll through the recent posts. Click on any post to view detailed information and interact. Follow/unfollow, upvote/downvote posts by clicking on the relevant field on their cards.
3) Search for Content: Use the search bar in the navigation bar to find specific items by title or use the advanced search button on the search field to conduct advanced search.
4) Profile Options: Click your profile icon to navigate to your user profile or other account-specific settings.

Logout: Click the Logout button to securely log out of your account.

4.7.5. Profile Page


Account created successfully!

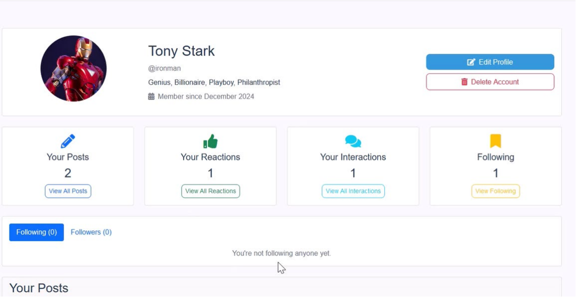
peter parker

@spiderman

Hello, I am a member of this platform.

Member since December 2024

You're not following anyone yet.


Tony Stark

@ironman

Genius, Billionaire, Playboy, Philanthropist

Member since December 2024

You're not following anyone yet.

Left: The Profile Page of a new user redirected from Sign Up Page.
Right: The Profile Page of a user redirected from Profile button:

4.7.5.1. Purpose

The Profile Page allows logged-in users to view and manage their account details, posts, reactions, interactions, and follow status. It provides options for editing the profile, deleting the account, and accessing specific activity metrics.

4.7.5.2. Key Features

4.7.5.2.1. User Information*: Displays the user's profile picture, username, bio, and membership information.*

4.7.5.2.2. Account Actions:

- Edit Profile: Opens the profile editing form where users can update their bio, profile picture, and other personal details.
- Delete Account: Permanently deletes the user's account and all associated data (requires confirmation).

4.7.5.2.3. Activity Metrics:

- Your Posts: Displays the number of posts created by the user. Includes a View All Posts button to access a detailed list of all posts.
- Your Reactions: Displays the number of reactions (votes) the user has made. Includes a View All Reactions button for detailed information.
- Your Interactions: Displays the number of interactions (e.g., comments) the user has made. Includes a View All Interactions button for further details.
- Following: Shows the number of users or posts the user is following. Includes a View Following button for detailed information.

4.7.5.2.4. Follow Status:

- Displays the number of users the profile owner is following and their followers.

4.7.5.2.5. Activity Details:

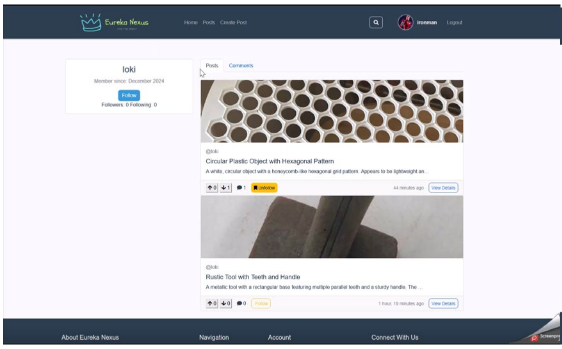
- Each card includes:
  - Image: A preview of the object.
  - Title and Description: A brief description of the object.
  - Posted By: The username of the contributor redirecting to the contributor's profile.
  - Metadata: Provides additional details on voting, comments.

4.7.5.3. How to Use

1) Edit Profile: Click the Edit Profile button to modify your profile information or update your profile picture. Save changes once edits are completed.
2) Delete Account: Click the Delete Account button to permanently remove your account. Enter the password and confirm the action when prompted. This action is irreversible, and all user data will be permanently removed. All the posts of the account become anonymous after the deletion.

3) View Metrics: Click View All Posts, View All Reactions, or View All Interactions or scroll down to explore detailed data about your activity.
4) Manage Follow Status: View the number of users you are following and your followers. Click View Following to see a detailed list.
5) Interact with Posts: Scroll down to the Your Posts section to view or interact with individual posts.

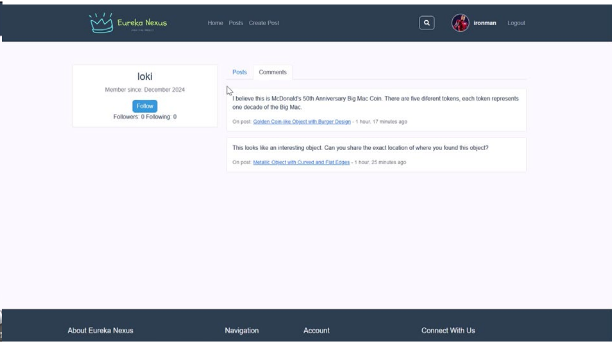
4.7.6. The Other User’s Profile Page



loki
Member since: December 2024
Followers: 0 Following: 0

@loki
Circular Plastic Object with Hexagonal Pattern
A white, circular object with a honeycomb-like hexagonal grid pattern. Appears to be lightweight an...
↑0 ↓1 1 — 44 minutes ago

@loki
Rustic Tool with Teeth and Handle
A metallic tool with a rectangular base featuring multiple parallel teeth and a sturdy handle. The ...
↑0 ↓0 0 — 1 hour, 19 minutes ago



loki
Member since: December 2024
Followers: 0 Following: 0

I believe this is McDonald's 50th Anniversary Big Mac Coin. There are five diferent tokens, each token represents one decade of the Big Mac.
On post: Golden Coin-like Object with Burger Design - 1 hour, 17 minutes ago

This looks like an interesting object. Can you share the exact location of where you found this object?
On post: Metallic Object with Curved and Flat Edges - 1 hour, 25 minutes ago

4.7.6.1. Purpose

The Other User's Profile Page allows users to view the posts and comments of another community member. This page is accessible by clicking on the username displayed on posts or post cards.

4.7.6.2. Key Features

4.7.6.2.1. User Information:

- Displays the selected user's username and membership details.
- Shows the user's follower count, including the number of followers and users they are following.

4.7.6.2.2. Follow/Unfollow Button:

- Allows you to follow or unfollow the user for updates on their activity.

4.7.6.2.3. Content Tabs:

- Posts Tab: Displays all posts created by the user.
- Comments Tab: Displays all comments made by the user (accessible by switching to the "Comments" tab).

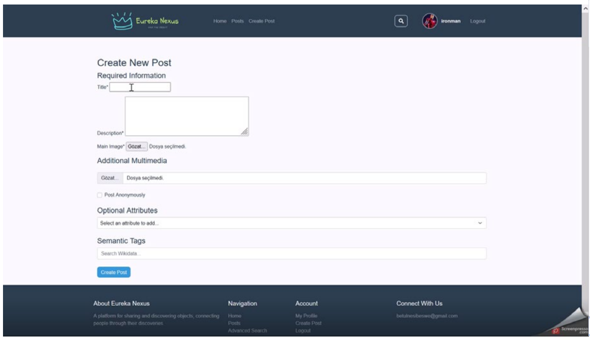
4.7.6.2.4. Post Previews:

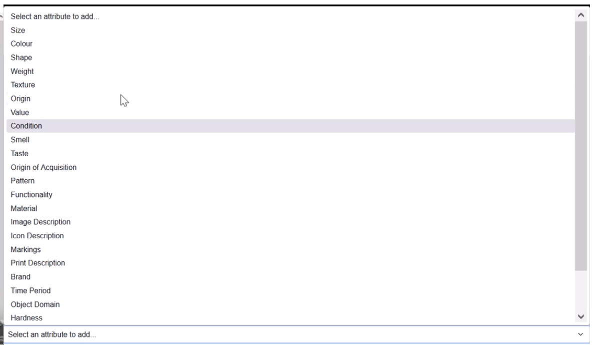
- Displays a list of the posts made by the user. Each card includes:
    - Image: A preview of the object.
    - Title and Description: A brief description of the object.
    - Posted By: The username of the contributor redirecting to the contributor’s profile.
    - Metadata: Provides additional details on voting, comments.
    - Follow Button: Indicates if the post is followed or not.

4.7.6.3. How to Use

1) Follow/Unfollow the User: Click the Follow button to start following the user. If you are already following the user, the button will display Unfollow. Click it to unfollow the user.
2) Explore User Content: Navigate between the Posts and Comments tabs to view the user's contributions. Click the View Details button on any post preview to see the full post.

4.7.7. Creating a Post





4.7.7.1. Purpose

The Create New Post Page allows logged-in users to share their objects with the Eureka Nexus community. Users can add detailed descriptions, multimedia, and semantic tags to enhance their posts.

4.7.7.2. Fields and Validation Rules

4.7.7.2.1. Required Fields

- Title: Provide a concise and descriptive title for your post.
- Description: Add a detailed description of the object.
- Main Image: Upload an image representing the object.
- Error messages are displayed below the respective fields if required fields are not filled.

4.7.7.2.2. Optional Fields

- Additional Multimedia: Upload additional images or files to provide more context about the discovery.
- Post Anonymously: Check this box to hide username on the post.
- Optional Attributes: Use the dropdown menu to select and add attributes such as size, weight, color, or other object characteristics. An optional attribute can be added as much as user wants. They can be removed by a remove button next to them.
- Semantic Tags: Use the Search Wikidata field to add semantic tags that enhance discoverability and context for the post.

4.7.7.3. How to Create a Post

1) Fill in the Title and Description fields.
2) Upload the Main Image.
3) Optionally, add Additional Multimedia and select Optional Attributes.
4) Enter any relevant Semantic Tags by searching Wikidata.
5) Check the Post Anonymously box if you do not wish to display your username.
6) Click the Create Post button to publish your post.

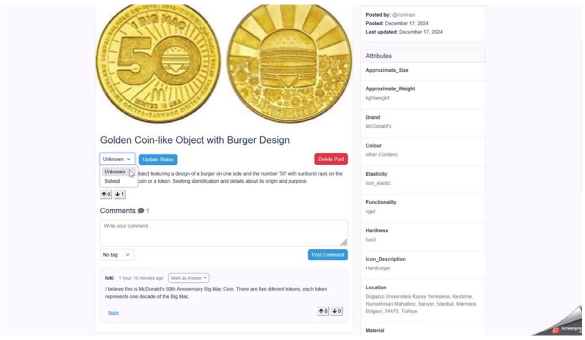
4.7.7.4. Tips for Creating a Successful Post

- Use a clear and descriptive title to attract attention.
- Provide detailed information in the description to help the community understand your object.
- Ensure the uploaded image is showing the object properly.
- Utilize semantic tags to make your post more searchable and semantic.

4.7.7.5. Error Messages

When clicked on Create Post, if a required field is not entered, the following error message will appear:

4.7.8. User’s Own Post Page



Golden Coin-like Object with Burger Design

bject featuring a design of a burger on one side and the number '50' with sunburst rays on the
coin or a token. Seeking identification and details about its origin and purpose.

Write your comment...

loki 1 hour, 16 minutes ago

I believe this is McDonald's 50th Anniversary Big Mac Coin. There are five diferent tokens, each token represents one decade of the Big Mac.

Posted by: @ironman
Posted: December 17, 2024
Last updated: December 17, 2024

Attributes

Approximate_Size

Approximate_Weight
lightweight

Brand
McDonald's

Colour
other (Golden)

Elasticity
non_elastic

Functionality
rigid

Hardness
hard

Icon_Description
Hamburger

Location
Boğaziçi Üniversitesi Kuzey Yerleşkesi, Kestirme, Rumelihisarı Mahallesi, Sarıyer, İstanbul, Marmara Bölgesi, 34470, Türkiye

Material

4.7.8.1. Purpose

The User's Own Post Page allows the post creator to manage their post, interact with comments, and update the post status. It provides options to tag comments as answers, update the post status, and delete the post if necessary.

4.7.8.2. Key Features

4.7.8.2.1. Post Information:

- Displays the post's title, description, and multimedia (e.g., images).
- Shows filled optional attributes, such as size, weight, brand, colour, and location.

4.7.8.2.2. Status Management:

- The post's status can be set to either:
  - Unknown: The default status when the post is created.
  - Solved: Indicates the object has been identified or resolved.
- Requirement for Solved Status: At least one comment must be tagged as an Answer before the post status can be updated to Solved.

4.7.8.2.3. Comment Management:

- Users can tag comments as Answer by selecting the "Mark as Answer" option on individual comments.
- Users can remove the Answer tag from comments if needed.

4.7.8.2.4. Post Actions:

- Update Status: Changes the post's status based on the user's selection (Unknown or Solved).
- Delete Post: Permanently removes the post and all associated comments.

4.7.8.2.5. Voting and Interaction:

- View the voting count for the post and interact with it using upvote/downvote buttons.
- Add new comments using the Post Comment field.
- Add replies using Reply link under comments.

4.7.8.3. How to Use

1) Update Post Status:
   a. Select a status from the dropdown (Unknown or Solved).
   b. To mark as Solved, ensure at least one comment is tagged as Answer.
   c. Click the Update Status button to save the change.
2) Manage Comments:
   a. To mark a comment as Answer, click the "Mark as Answer" option next to the comment.
   b. To remove the Answer tag, click the “Remove the Answer” option next to the comment.
3) Post a Comment:
   a. Enter the comment in the Write your comment... field.
   b. Optionally, select a tag “Hint” or “Question” for the comment.

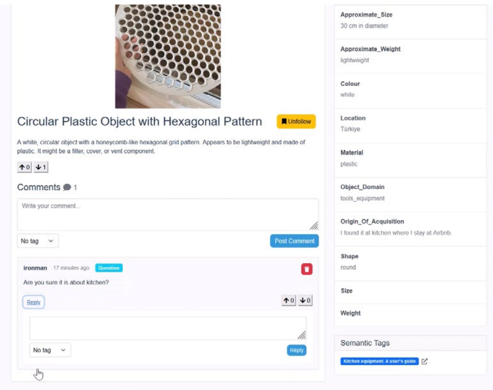
c. Click the Post Comment button to add the input.

4) Post a Reply:
   a. Click the Reply link to open reply text field.
   b. Click the Post Comment button to add your input.
5) Delete Comment:
   a. If it is wished to remove the comment, click the trash icon next to the comment. Confirm the action when prompted.
6) Delete Post:
   a. If it is wished to remove the post, click the Delete Post button. Confirm the action when prompted.

4.7.8.4. Error Messages

When clicked on Update Status, if none of the comments is selected as Answer, the following error message will appear: "Cannot mark as solved: No comment has been marked as an answer yet."

4.7.9. Other User’s Post Page



Circular Plastic Object with Hexagonal Pattern

A white, circular object with a honeycomb-like hexagonal grid pattern. Appears to be lightweight and made of plastic. It might be a filter, cover, or vent component.

↑0 ↓1

Comments 1

ironman 17 minutes ago Question

Are you sure it is about kitchen?

↑0 ↓0

Approximate_Size
30 cm in diameter

Approximate_Weight
lightweight

Colour
white

Location
Türkiye

Material
plastic

Object_Domain
tools_equipment

Origin_Of_Acquisition
I found it at kitchen where I stay at Airbnb

Shape
round

Size

Weight

Semantic Tags

Kitchen equipment, A user's guide

4.7.9.1. Purpose

The Post Page allows the user to vote and interact with comments and the post.

4.7.9.2. Key Features

4.7.9.2.1. Post Information:

- Displays the post's title, description, and multimedia (e.g., images).
- Shows filled optional attributes, such as size, weight, brand, colour, and location.

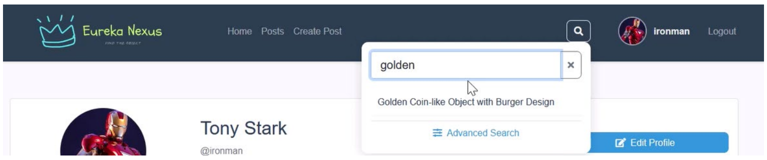
4.7.9.2.2. Voting and Interaction:

- View the voting count for the post and interact with it using upvote/downvote buttons.
- Add new comments using the Post Comment field.
- Add replies using Reply link under comments.

4.7.9.3. How to Use

1) Post a Comment:
   a. Enter the comment in the Write your comment... field.
   b. Optionally, select a tag “Hint” or “Question” for the comment.
   c. Click the Post Comment button to add the input.
2) Post a Reply:
   a. Click the Reply link to open reply text field.
   b. Click the Post Comment button to add your input.
3) Delete Comment:
   a. If it is wished to remove the comment, click the trash icon next to the comment. Confirm the action when prompted.

4.7.10. Searching an Object (Basic Search)



golden

Golden Coin-like Object with Burger Design

Tony Stark

@ironman

4.7.10.1. Purpose

The Basic Search allows users to quickly find posts on the platform by searching for keywords in the post titles. This feature is accessible from the search bar located in the navigation bar.

4.7.10.2. How to Use

1) Access the Search Bar: Click the search icon in the navigation bar to expand the search input field.
2) Enter Keywords: Type a keyword or phrase in the search bar that corresponds to the post title you are looking for.
3) View Search Results: Relevant matches will be displayed as a dropdown list below the search bar. Click on a result to navigate directly to the post.

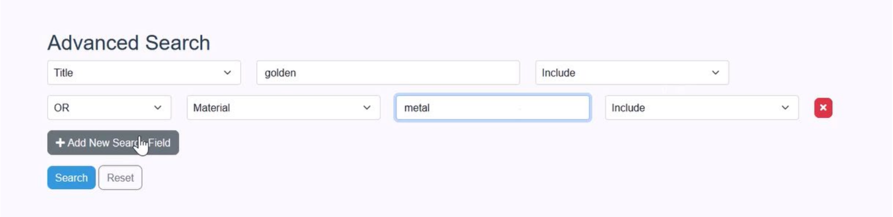
4.7.10.3. Key Features

4.7.10.3.1. Instant Search Results: *Matches are displayed dynamically as you type.*

4.7.10.3.2. Focused Search: *Searches are limited to titles, making it quick and efficient.*

4.7.10.3.3. Advanced Search Integration: *Users can easily switch to advanced search for more complex queries.*

4.7.11. Searching an Object (Advanced Search)



Title golden Include

OR Material metal Include

4.7.11.1. Purpose

The Advanced Search feature provides users with a flexible and detailed way to find posts by applying multiple filters and conditions. This allows for more precise searches compared to the Basic Search.

4.7.11.2. How to Use

1) Access Advanced Search: Click the Advanced Search link from the Basic Search dropdown or navigate directly to the Advanced Search page.
2) Define Search Criteria: Select a field to search from the dropdown menu. Enter a keyword or value in the corresponding field. Choose whether to search for Include or Exact Match the specified value using the dropdown menu.
3) Add Additional Conditions: Click the Add New Search Field button to include more criteria. Use the AND/OR/DO NOT INCLUDE toggle to define the logical relationship between multiple search fields.

4.7.11.3. Key Features

4.7.11.3.1. Multiple Filters: Combine conditions across attributes of a post.

4.7.11.3.2. Logical Operators: Use AND for narrowing searches, OR for broadening them, and DO NOT INCLUDE for avoiding them.

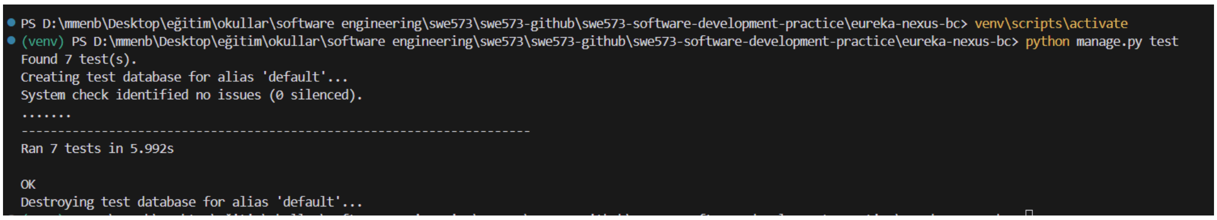
4.7.11.3.3. Include/Exact Match Options: Specify whether to get included or exact match of the keywords or values in your search.

4.7.11.3.4. Dynamic Field Addition: Add as many fields as needed to refine your search.

4.8. Test results (unit & user)

4.8.1. Unit Test Results

As stated on 4.4. Project Status, only account registration management and user management tests are created. When these tests are run, they return “OK” as below.



```
● PS D:\mmenb\Desktop\eğitim\okullar\software engineering\swe573\swe573-github\swe573-software-development-practice\eureka-nexus-bc> venv\scripts\activate
● (venv) PS D:\mmenb\Desktop\eğitim\okullar\software engineering\swe573\swe573-github\swe573-software-development-practice\eureka-nexus-bc> python manage.py test
  Found 7 test(s).
  Creating test database for alias 'default'...
  System check identified no issues (0 silenced).
  .......
  ----------------------------------------------------------------------
  Ran 7 tests in 5.992s

  OK
  Destroying test database for alias 'default'...
```

4.8.2. User Test Results

User testing was conducted to evaluate the core functionalities of the platform, including login, signup, profile management, voting, commenting, following, and posting. The results shows that these primary features of the platform perform reliably and meet the expected requirements of the project. Users were able to register successfully, log in without errors, and manage their profiles, including editing profile details and navigating to their personalized dashboard. The functionality for creating, updating, and interacting with posts worked well, with users able to submit posts, add multimedia content, and utilize the voting system effectively. Similarly, the commenting feature allowed threaded interaction, enabling users to provide feedback or engage in discussions under posts. The following functionality, both for users and posts, operated correctly, ensuring proper tracking and notifications for interactions.

Despite these, the advanced search functionality, while operational, did not meet the expected levels of accuracy and precision. It requires further work on its functionality. Additionally, it is seen that there is no response mechanism for page errors like 404, etc.