SWE 573
SOFTWARE DEVELOPMENT PRACTICE

CONNECT THE DOTS

AUTHOR : YASEMİN TANGÜL

Overivew

Connect the Dots is an interactive knowledge graph platform that allows users to create meaningful structures by visually connecting pieces of information (nodes). The main goal of the project is to provide an intuitive and interactive environment where both individuals and researchers can organize their thoughts, concepts or content in a way that is related to each other.

Each node on the platform can be defined manually or created by automatically extracting information from Wikidata. Connections between nodes express relationships between concepts and can be defined by users. Thus, users not only enter information, but also visualize this information and gain a deeper understanding.

The system developed within the scope of the project offers a Django-based backend, PostgreSQL database, vis.js-based dynamic visualization infrastructure and a user-friendly interface. Bringing a creative approach to knowledge management processes, Connect the Dots aims to increase individual productivity and present knowledge more efficiently by associating it.

Software Requirements Specification
Detailed SRS Dcoumentation: SRS - Github

- Requirements
    - Functional Requirements
      The system must allow users to create, explore and analyze connections between different entities.
      Users must be able to interact with the system by searching, browsing, adding content, and analyzing data.
      Users must be able to edit, annotate, and expand existing links. A change history should be maintained for changes made.
      The system must support both user-based and content-based connections.
      Users must be able to determine whether content is useful or not through a voting system.
      Predefined labels must be provided for users.
      Graphical representation must be used and connections should be filterable.
      Users must be able to see the topics they have contributed to.
      There must be a user scoring system based on user contributions and votes, but this will not be included in the first version.
      Users must register and anonymous logins are not allowed.
      Users must add their own business and location information.
    - Non-Functional Requirements
      The product must only provide English support.
      The user interface must be clear and easy to use.
      The product must be web application.

Categories must not be used, content classification must be done only through tags.
Adding AI is not mandatory, but the system must work with Wikidata.
There will be no real-time chat feature added, but there must be a discussion section.
Popular and trending topics must be displayed based on user activity.
If the contents are too similar, it is suggested to merge them.
There must be a contributors section for each topic.
Media integration must be provided and fast links must be offered to users.
There must be more than one board in the system and each board must have an owner.
The system must provide guidance to users on generating content from Wikidata.
User verification will not be required as part of security requirements, but anonymous logins will not be allowed.

- Product Constraints
    - Language Support: The application will only offer English language support. Other languages will not be supported.
    -
    - Anonymous Login is Prohibited: Users are required to register to access the system. Anonymous (guest) logins are not allowed.
    -
    - Category System Will Not Be Used: Content will be classified only through tags instead of being divided into categories.
    -
    - Artificial Intelligence Integration is Not Mandatory: The system does not have to work with artificial intelligence support. However, the system must be compatible with Wikidata.
    -
    - No Real-Time Chat: There will be no real-time messaging feature in the application. Instead, a discussion area will be provided for each topic.
    -
    - User Verification is Not Mandatory: Registered users do not need to verify their identities. However, users are required to provide their work and location information.
    -
    - No Point System in the First Version: A point system is planned according to user contributions, but this feature will not be included in the first version.
    -
    - Must be a Web-Based Application: The product will be developed to be accessible only via the web. Mobile or desktop application versions are not planned.

- User Characteristics
    - The target user group of this system consists of individuals who are willing to produce and share information, have analytical thinking skills, are researcher and accustomed to using interactive platforms. Users are expected to have the following characteristics:
    -
    - Registered Users: Access to the system is only possible for registered users. Anonymous or guest logins are not allowed.
    -
    - Professional and Individual Participants: Users are responsible for entering their own work (profession) and location information when producing content in the system. Therefore, the system targets both individual users and people with professional backgrounds.
    -
    - Ability to Produce and Interpret Content: Users are expected to be competent in establishing connections between entities, adding explanations to these connections and creating meaningful content.
    -
    - Familiarity with Technology: Since a web-based interface is used, it is sufficient for users to have basic computer and internet knowledge. The user-friendly interface minimizes the need for technical knowledge.
    -
    - Ability to Provide Interactive Contributions: Users can contribute to content, vote, comment on others' contributions, and share ideas in discussion areas.
    -
    - Contribution-Focused Profile Development: Users can track the topics they contribute to and create a profile over time by evaluating their contribution history in the system.

- Product Functions
    - This system provides a platform that allows users to establish, organize, and analyze connections between various entities. The basic functions are:

    - Link Creation and Discovery: Users can create new connections between different contents (people, places, concepts, etc.) and visually examine existing connections.

    - Search and Browse: Users can explore content in the system by searching or browsing through tags.

    - Add and Edit Content: Users can add new content, add descriptions/tags to existing links, and edit them. All changes are recorded in the change history.

- Voting System: Users can vote on the usefulness of the content; these votes are effective in highlighting the content.

- Tagging System: Content is classified only with tags. Categories are not used. The system offers predefined tags to the user.

- Graphical Display: Links are displayed in a graphical interface. Users can filter the links.

- Contribution Tracking: Every user can view the topics they have contributed to. Contributors for each topic are listed in a separate section.

- Discussion Area: Although there is no real-time messaging, there is a discussion section where users can share their opinions on every topic.

- Popular and Trending Topics: The system lists the trending and trending topics according to user activity.

- Content Merge Suggestion: If two contents are very similar, the system suggests merging them.

- User Registration and Profile Information: Anonymous login is not allowed. Users must add their work and location information when registering.

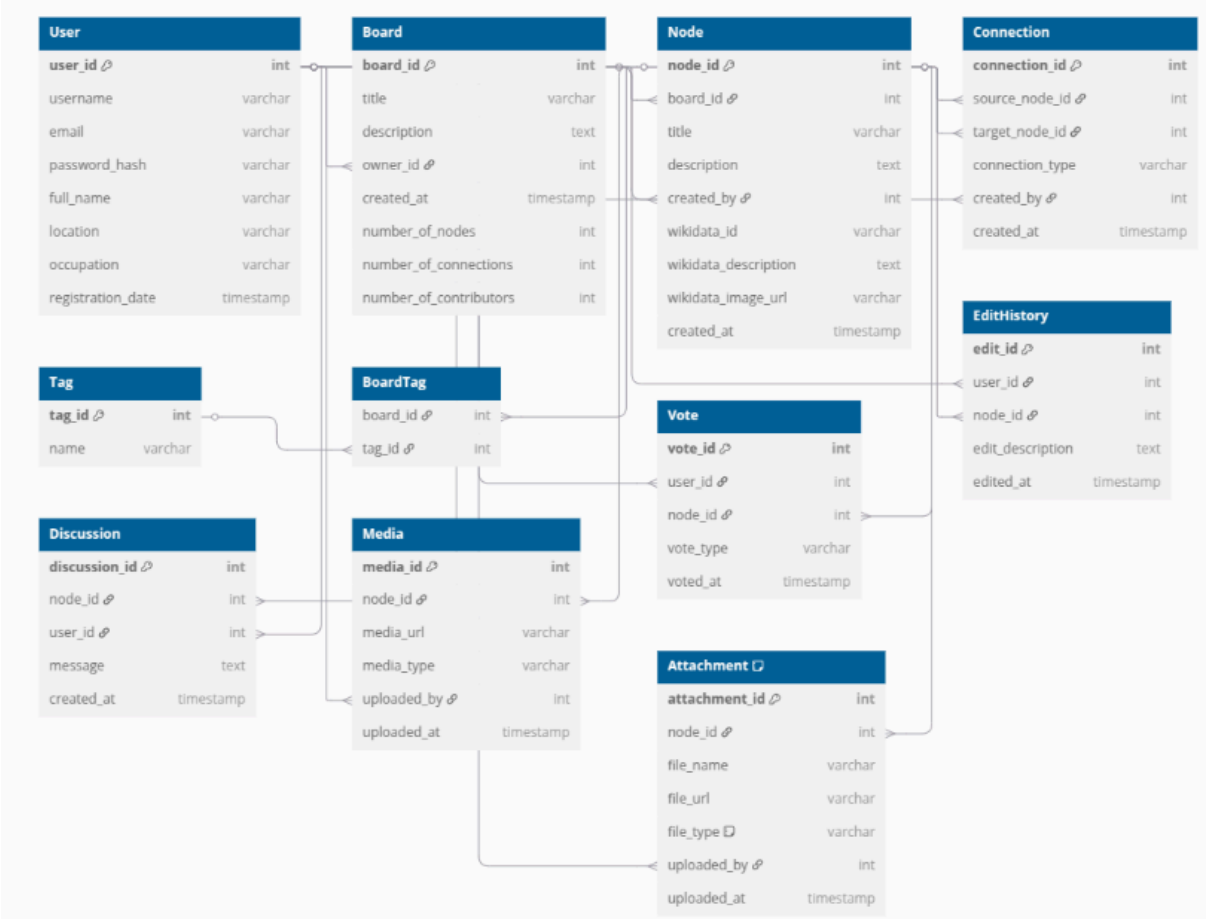
- Board Management: There are multiple boards in the system. Each board has an owner. These boards allow the links to be organized.

- Media and Quick Link Integration: Users can add media files to the content and use quick access links.

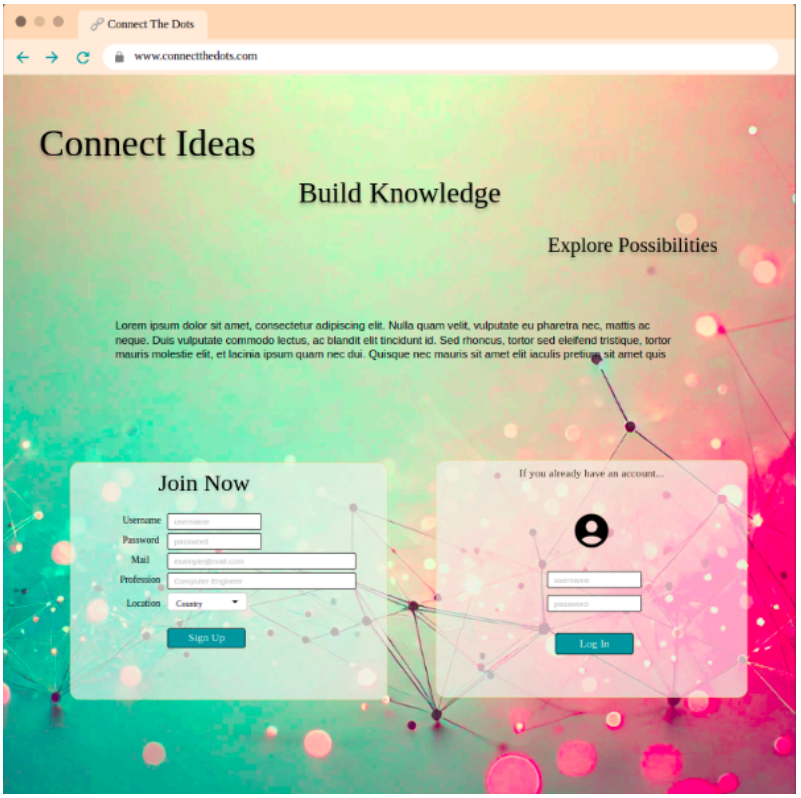
- Content Suggestion with Wikidata: The system works in conjunction with Wikidata to assist the user during content production.

Design Documents

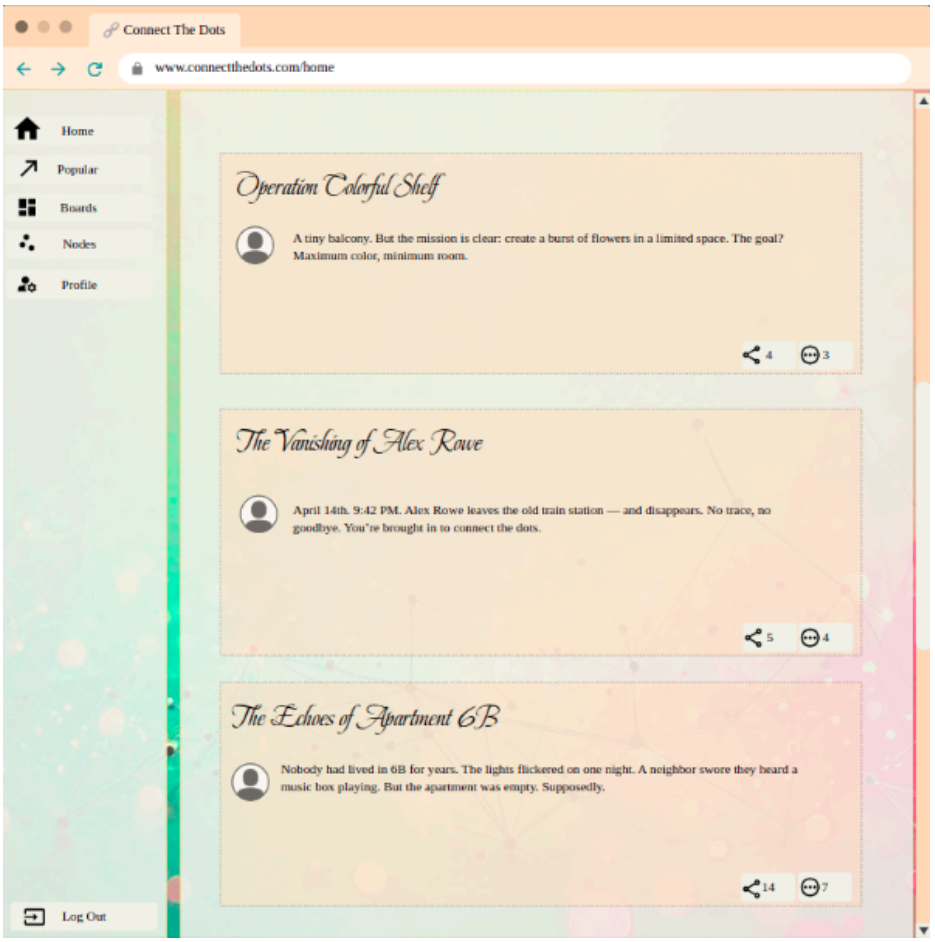
- **Database Design**



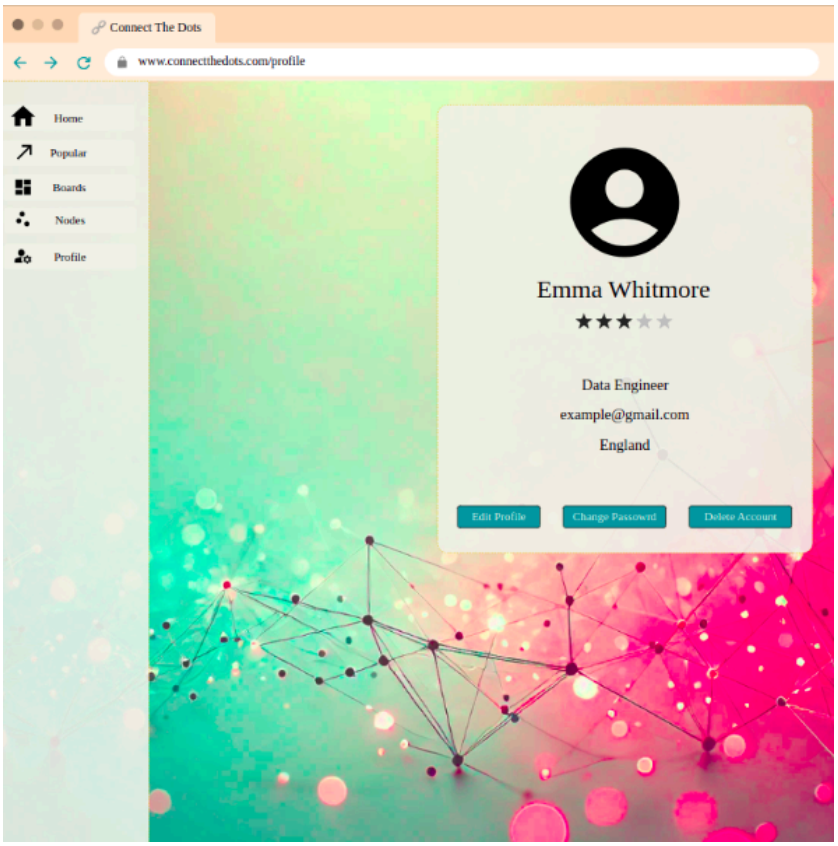
- **Mockups**
  Landing Page



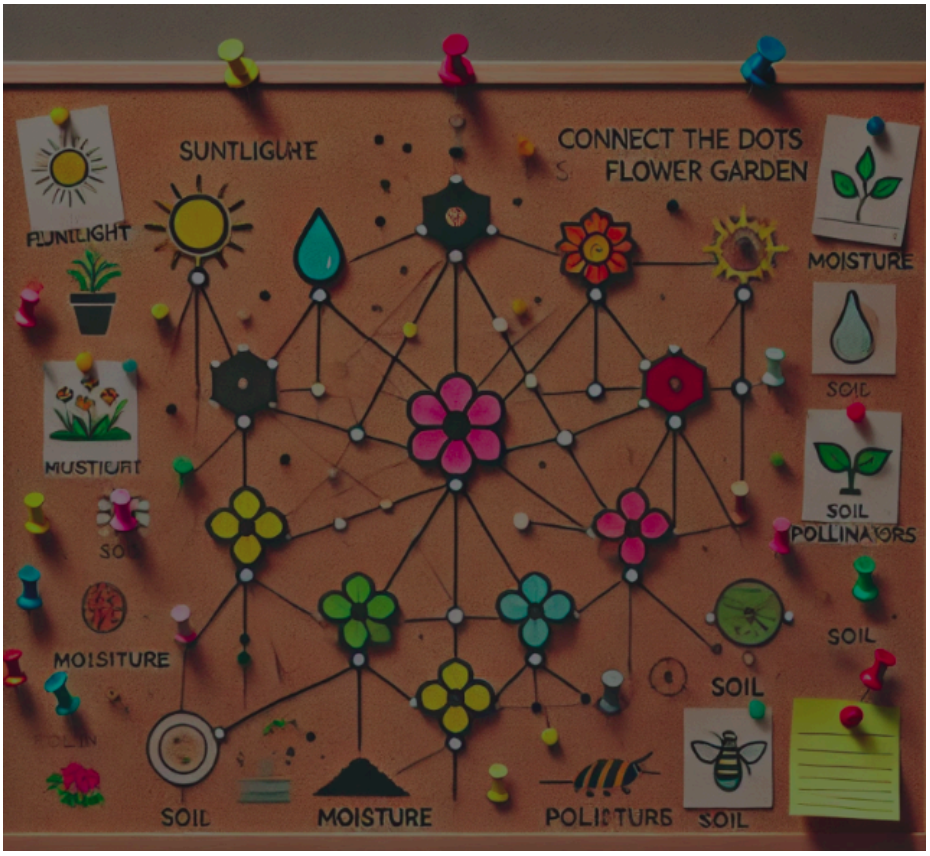
Homepage



Profile Page

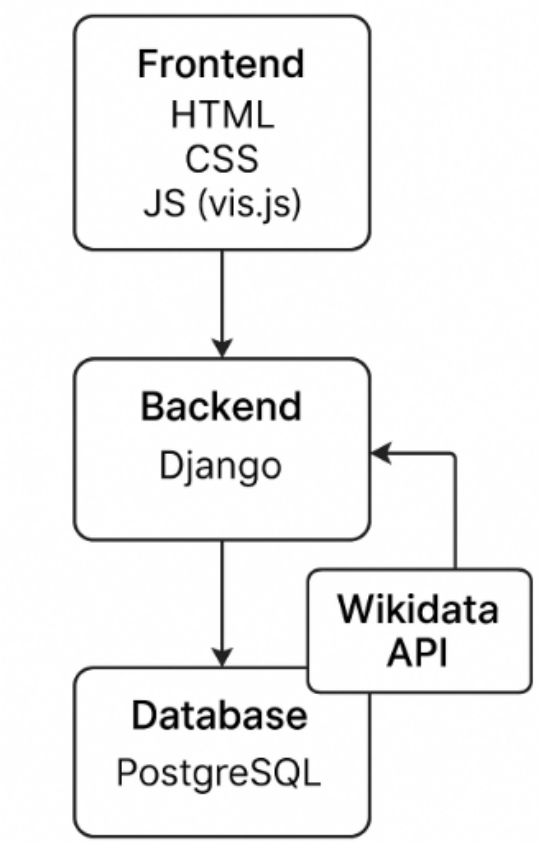


Inspiration for Detailed Board Page



Detailed Board Page

System Architecture



- Status of the Project

Requirement	Status
The system must allow users to create, explore and analyze connections between different entities.	Completed
Users must be able to interact with the system by searching, browsing, adding content, and analyzing data.	Partially completed (adding & browsing implemented, search/analysis missing)
Users must be able to edit, annotate, and expand existing links. A change history should be maintained for changes made.	Partially completed (editing works, history not implemented)
Users must be able to determine whether content is useful or not through a voting system.	Not completed
Predefined labels must be provided for users.	Not completed

Graphical representation must be used and connections should be filterable.	Partially completed (graph visualization completed, filtering not completed)
Users must be able to see the topics they have contributed to.	Completed
There must be a user scoring system based on user contributions and votes, but this will not be included in the first version.	Out of scope for initial version
Users must register and anonymous logins are not allowed.	Completed
Users must add their own business and location information.	Completed
The product must only provide English support.	Completed
The user interface must be clear and easy to use.	Completed
The product must be a web application.	Completed
Categories must not be used, content classification must be done only through tags.	Not completed
Adding AI is not mandatory, but the system must work with Wikidata.	Completed
There will be no real-time chat feature added, but there must be a discussion section.	Not completed
Popular and trending topics must be displayed based on user activity.	Not completed
If the contents are too similar, it is suggested to merge them.	Not completed
There must be a contributors section for each topic.	Not completed
Media integration must be provided and fast links must be offered to users.	Not completed
There must be more than one board in the system and each board must have an owner.	Completed

The system must provide guidance to users on generating content from Wikidata.	Not Completed
User verification will not be required as part of security requirements, but anonymous logins will not be allowed.	Completed

- Status of Deployment
  Dockerized:
  Dockerfile:GIT- Dockerfile
  Docker Compose: GIT - Docker Compose

  Deployed on AWS: http://13.60.196.130:8000

- Installation

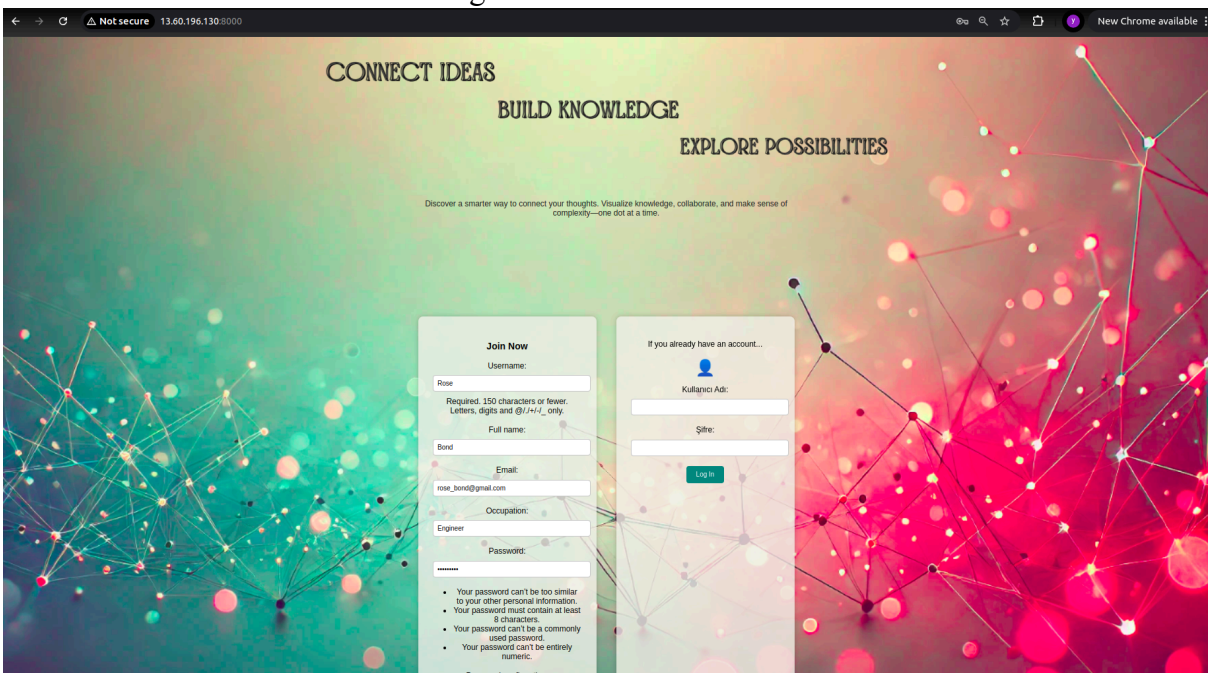
  From Repository:

  >> git pull https://github.com/tyasemin/SWE573.git
  >> Create .env file (Check the docker-compose yml for the arguments)
  >> docker build -t <image_name> -f Dockerfile<or dockerfile_path> .
  >> Replace the image name in the docker-compose.yml
  >>docker compose (or docker-compose depending on your system) up

- User Manual
  test user name: Rose
  user last name: Bond
  user gmail: rose_bond@gmail.com
  occupation: Engineer
  password: A78561278

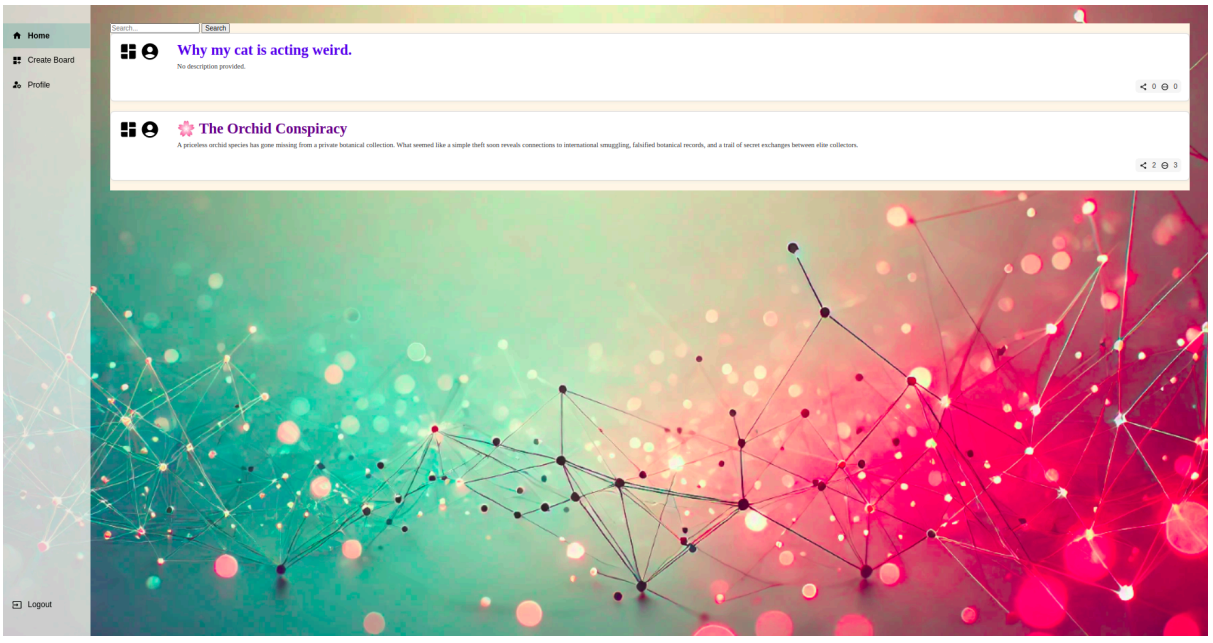
Landing Page:

Fill the above information to user registration form.



Home Page:

When you register the system, it will redirect you to the home page, you can see previously created boards from other users(and yours as well). Click the board to see the description, nodes in the board or you can search a certain topic (for board and description) from the search bar.



Detailed Board Page:

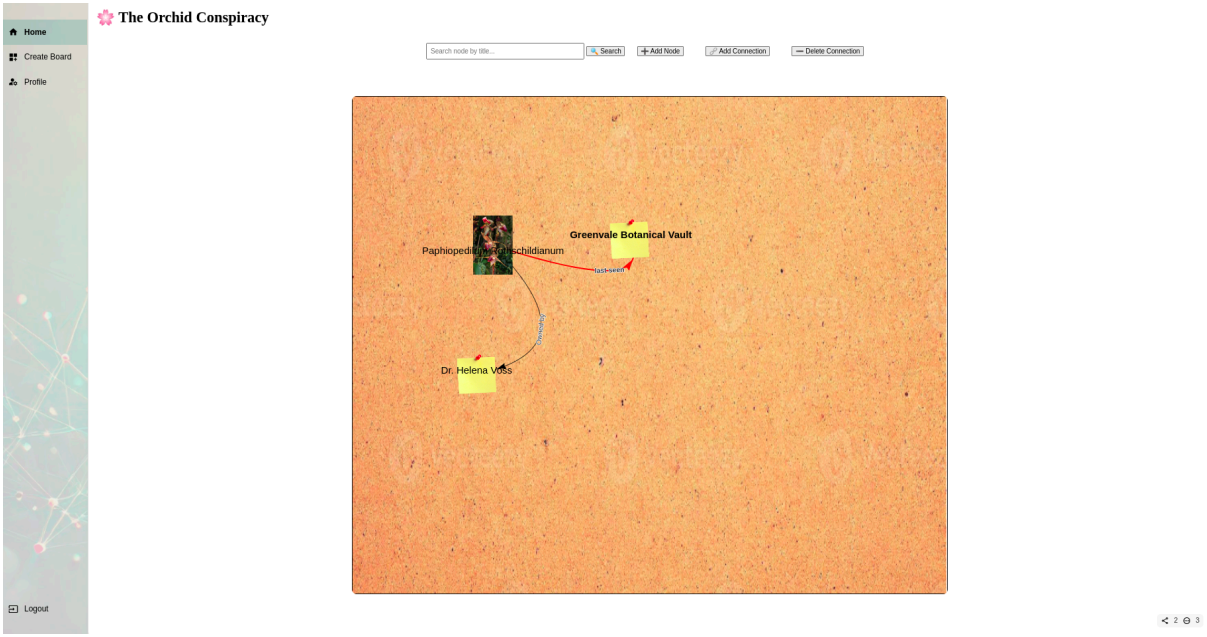
Double click the nodes to see the detailed information in the board.

“+ Add Node”, click to add new node to the board

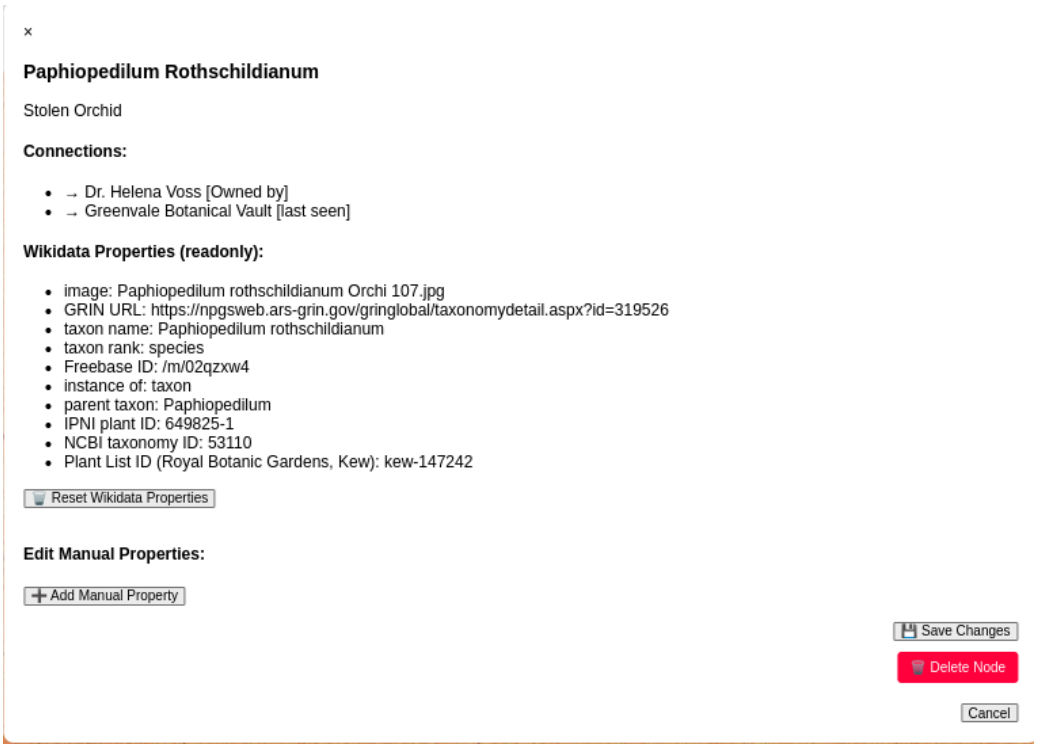
“Add Connection”, click to add new connection between nodes.

“Delete Connection”, click to delete connection between nodes.

You can change the placement of the nodes.



Double Click the Node:

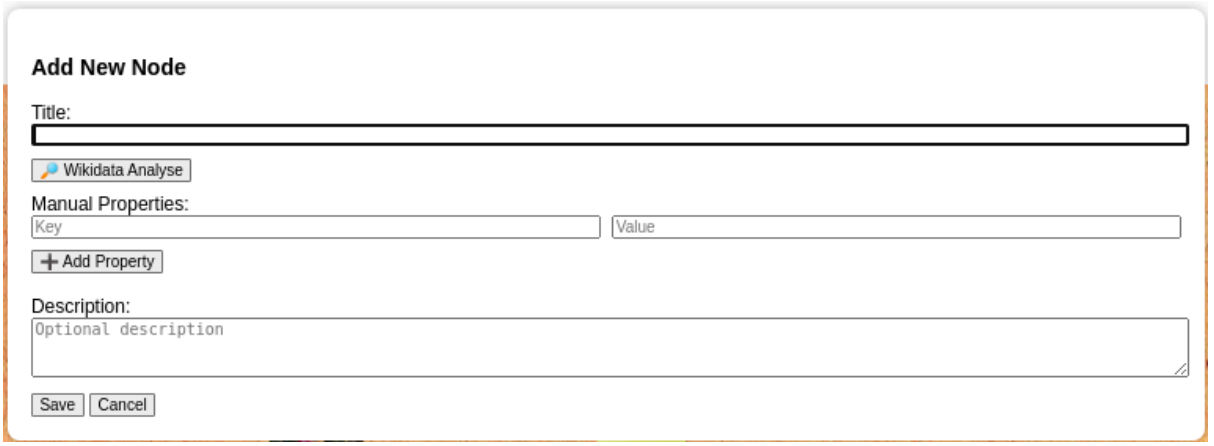


In this node you can reset the wikidata properties but creating a new wikidata analyze is forbidden. Or you can add/edit manual added properties.

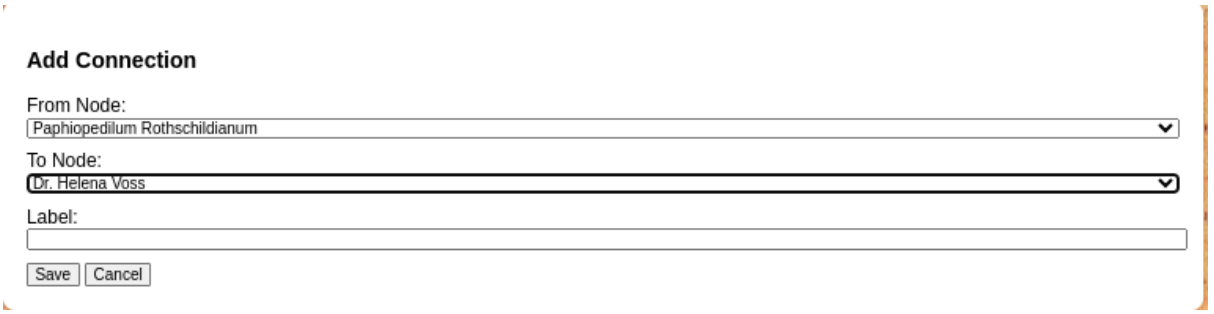
Click to Add Node:

Fill the empty fields.

If you like the add wikidata properties, click Analyze Wikidata, otherwise fill the properties by hand.

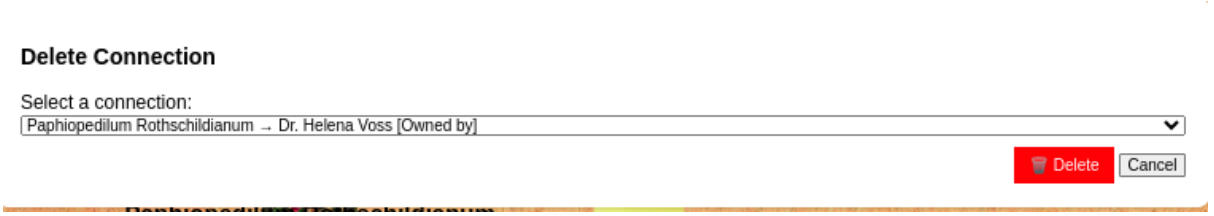


Click to Add Connection:



Select the from node, to node and give a label to show to connection of the nodes.

Click Delete Connection



Select the connection you like to delete.

- In the current version of the project:

- CI Tests: A basic continuous integration (CI) mechanism is in place. During the Docker image build phase, the system verifies whether the Django backend can successfully connect to the PostgreSQL database. If the database connection fails, the

image is not published on the host environment. This acts as a basic health check for database integration.

- Code-level Unit Tests:
    - No formal unit tests have been implemented for backend models, views, or utility functions at this stage. Future iterations of the project aim to include automated unit testing using pytest or Django's built-in test framework to ensure stability and regression protection.

- During User Test,Key observations include:

    - Node and Connection Deletion Bug:
      When deleting nodes or their related connections, an error message such as "Connection failed" occasionally appears. However, this issue appears to be superficial, as refreshing the page resolves the inconsistency. It is likely due to the frontend not being updated after asynchronous deletion.

    - Modal Interaction:
      Users were able to create and edit nodes successfully using the modals. However, feedback indicated that more visible success/error messages and better state synchronization (especially after edits) would improve usability.

    - Wikidata Fetching:
      Users tested the Wikidata-based node creation flow. Most cases worked as expected, but some failed silently due to incomplete or ambiguous titles. Error handling for empty results is noted as an area for improvement.