**Dynamic web applications Portfolio Project – HiveFocus**

**Outline**

Hive Focus is an education-based productivity web application designed to help students stay focused, study effectively, and collaborate. The app features a Pomodoro-style timer that gamifies focus sessions, allowing users to earn points for completing study intervals. These points can be used to grow a virtual forest by purchasing virtual flowers and trees, providing a sense of progress and accomplishment.  
For every hour studied, Hive Focus contributes toward a donation supporting ethical honey farming, promoting sustainability and environmental responsibility. The web app also includes a student discussion forum where users can ask questions, share knowledge, and interact with peers. With its mission to improve focus while fostering ecological impact, Hive Focus embodies its slogan, "Reap What You Sow."

**Links**

* **Deployed App**: http://www.doc.gold.ac.uk/usr/175/
* **GitHub Repository**: https://github.com/Abida-Tarafdar/my-forum

**Architecture**

**High-Level Diagram:**

Hive Focus employs a two-tier architecture:

1. **Application Tier**: Built with Node.js and Express for backend functionality, EJS for server-side rendering, and HTML, CSS, and JavaScript for the frontend.
2. **Data Tier**: MySQL database manages user data (first name, last name, email), published posts and replies.

**Data Model Diagram:**

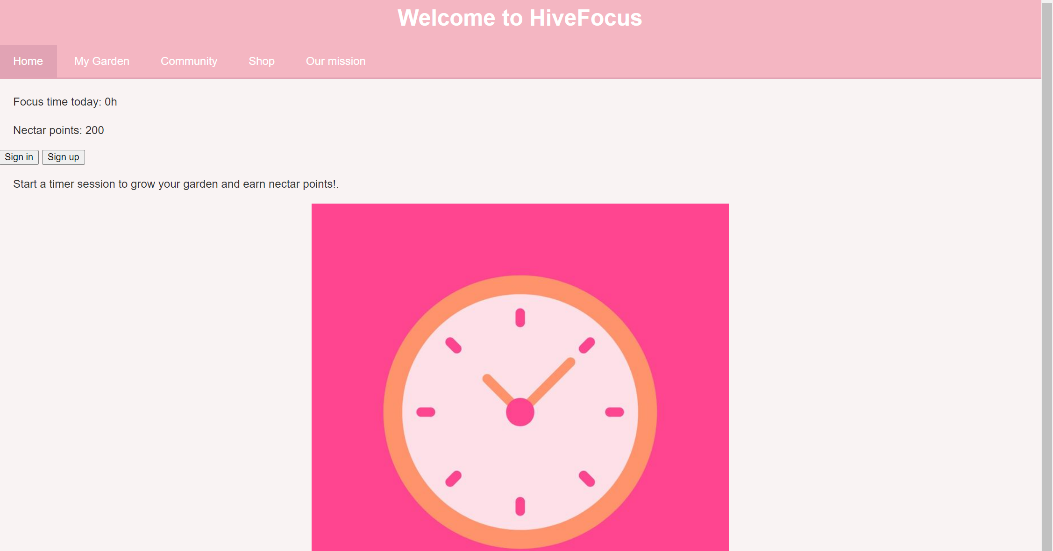
Hive Focus uses a relational data model:

* **Users Table**: Stores user credentials, profile data, points earned, and forest progress.
* **Posts Table**: Handles discussion forum posts and metadata.
* **Replies Table**: Tracks replies to posts.

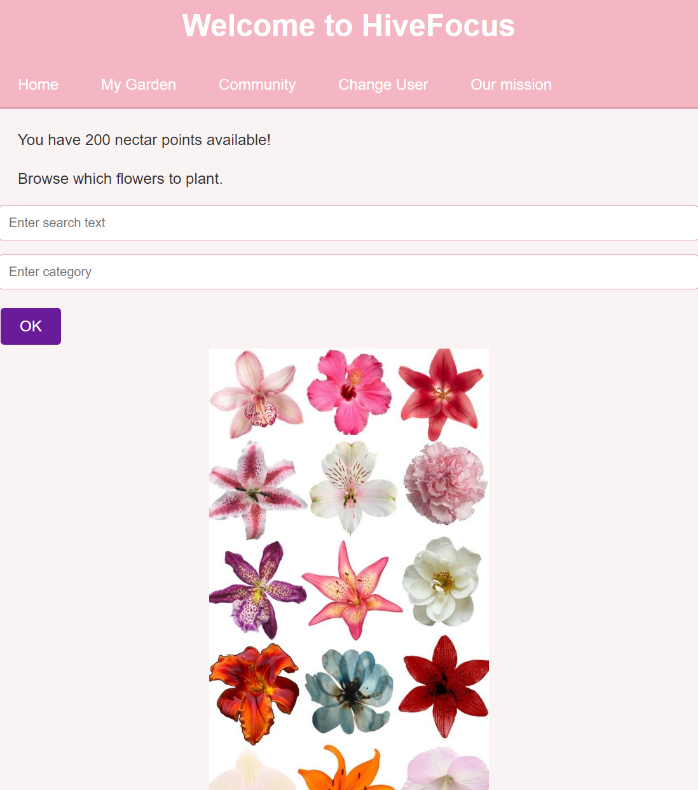
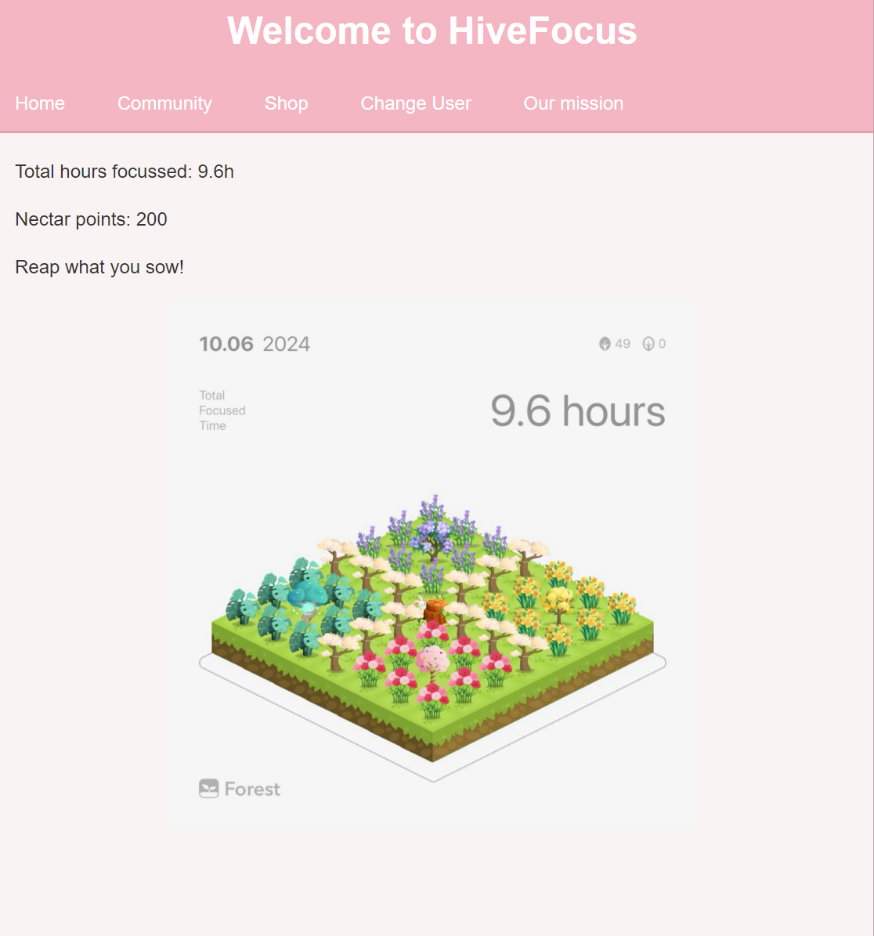
**User Functionality**

Hive Focus offers the following features to users:

**1. Pomodoro Timer with Gamified Progress**

* **Description**: A timer that helps users focus for intervals of 25 minutes with a 5-minute break. Completing intervals awards points, which contribute to both virtual rewards and real-world donations.
* **Screenshots**:
  + Timer interface. 

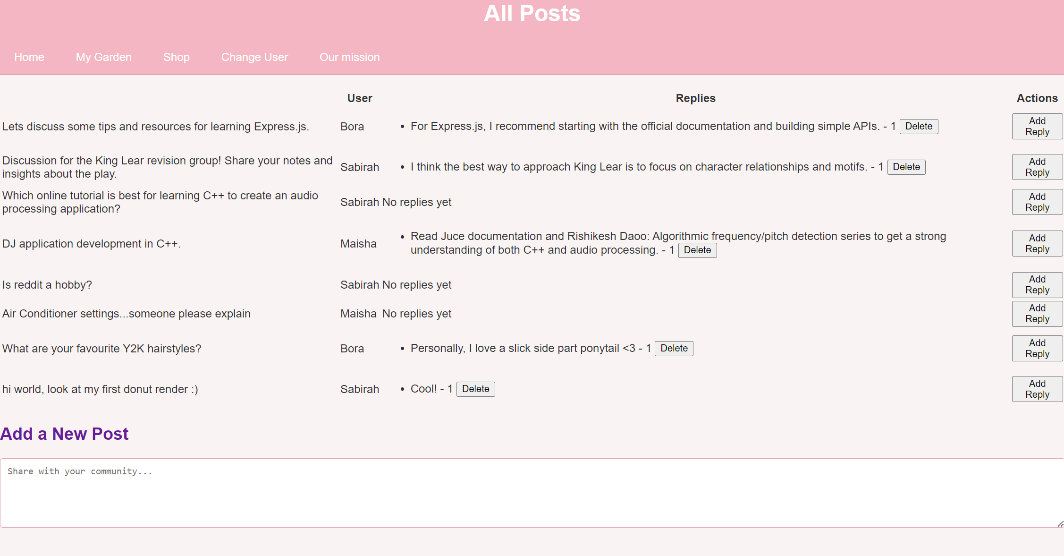
**2. Virtual Forest**

* **Description**: Users can spend earned points to grow a personalized virtual forest by purchasing flowers and trees.
* **Features**:
  + **Shop Interface**: Buy virtual plants using earned points.
  + **Forest Visualization**: See the plants grow in an interactive interface.
  + **Leaderboard**: Compare forests with other users.
* **Screenshots**:
  + Shop interface for buying plants.
  + 
  + A user’s virtual forest in progress.
  + 

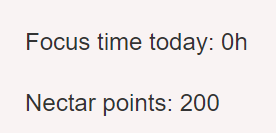
**3. Discussion Forum**

* **Description**: A collaborative space for students to discuss academic topics and share knowledge.
* **Features**:
  + Post and reply functionality.
  + Edit and delete posts or replies.

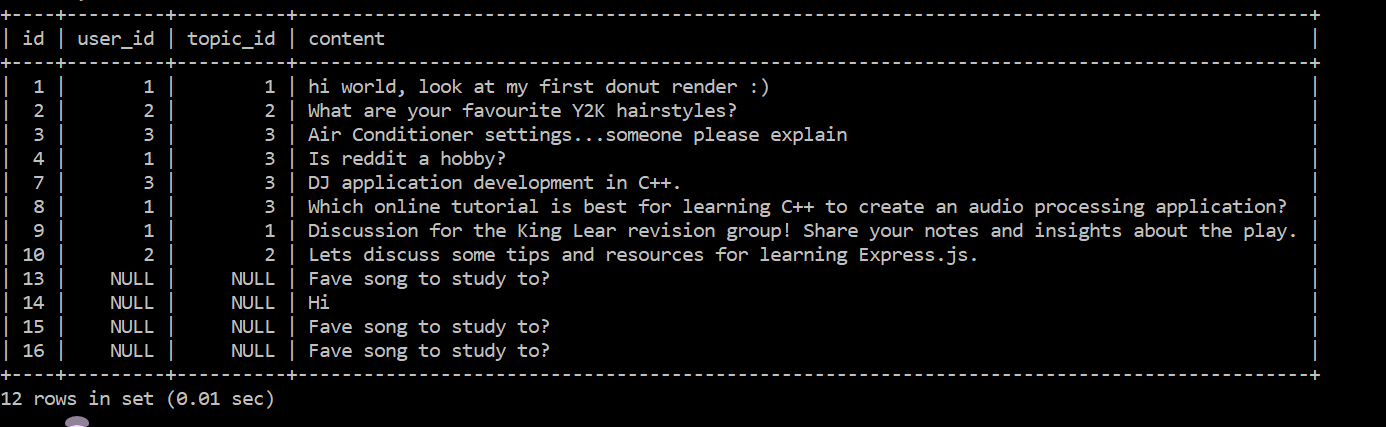
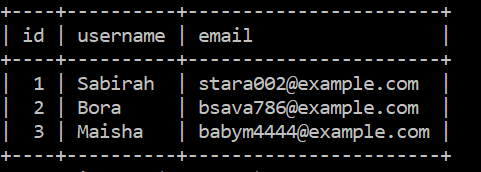
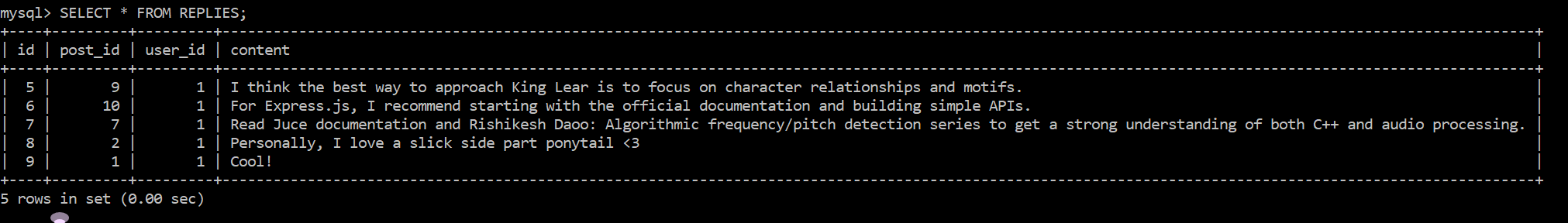
**Screenshots**:



**4. User Dashboard**

* **Description**: Displays user progress, including total points and hours studied.
* **Screenshots**:
  + Dashboard summary with donation and forest statistics. 

**SQL tables;**



The database structure establishes a set of clear relationships between tables through primary and foreign keys. The users table serves as a core reference, with its primary key (id) acting as a foreign key in other tables, such as posts, replies, and flashcards, to indicate ownership or authorship. The posts table connects to the users table via the user\_id foreign key and to the topics table through the topic\_id foreign key, linking each post to its creator and topic. Similarly, the replies table references posts via the post\_id foreign key and users via the user\_id foreign key, associating replies with their respective posts and authors. The topics table, while self-contained with a primary key (id), is linked to posts through the topic\_id foreign key. The flashcards table ties its entries to users using the user\_id foreign key, representing ownership. These relationships primarily exhibit one-to-many dynamics, such as one user having multiple posts, replies, or flashcards, and one topic encompassing multiple posts. Conversely, the many-to-one relationships indicate that each post, reply, or flashcard is linked back to a single user, ensuring referential integrity and enabling robust interaction across the tables.