Technical Document

Database Setup

My database was setup using php my admin. My SQL database was created named ‘trendtrove’. After I added the tables to manage users, pins, comments, notifications, messages, boards, followers, likes, analytics, board pins.

Two example of tables structure

The user table Structure:
    - `user_id` (Primary Key): Unique identifier for each user.
    - `username`, `email`, `password_hash`, `profile_picture`, `cover_picture`, `bio`

Fields to store user information.
  - **Pin Table:**
    - `PinID` (Primary Key): Unique identifier for each pin.
    - `UserID` (Foreign Key): References the user who created the pin.
    - `ImageURL`, `Description`, `season`, `CreationDate`: Fields to store pin details.
  - **Board Table:**
    - `BoardID` (Primary Key): Unique identifier for each board.
    - `UserID` (Foreign Key): References the user who created the board.
    - `BoardName`, `Description`: Fields to store board details.

Other tables such as: Comments, Notifications, Messages, Followers, Likes and Analytics have similar structures with their respective fields and foreign keys.

Relationships:
  - Each pin is associated with a user.
  - Each comment is associated with a user and a pin.
  - Each notification is associated with a user.
  - Each message is associated with sender and receiver users.
  - Each board pin links a board and a pin.
  - Followers and likes are associated with users and pins.

After I have created the folder in visual studio code and create the database connection file which there I have connected the database to make my website function.

Explain Techniques Used to Manipulate Data in Your Database Through Your Web Application

I have used the CRUD (Create, Read, Update, Delete) operations which are the fundamental techniques I have used to manipulate data in the database In the Trendtrove application. These operations are implemented using PHP and MySQL to handle interactions between the web application and the database.

- Create:
  User registration and pin creation involve inserting new records into the `users` and `pins` tables respectively.

  - Read:
Displaying pins, comments, user profiles, and notifications involves selecting data from the relevant tables.

  - Update:
  Editing user profiles or pin details involves updating existing records in the `users` and `pins` tables.

  - Delete:
  Removing pins or comments involves deleting records from the `pins` and `comments` tables.

Prepared Statements:
  - Used to prevent SQL injection attacks.
  - Example: `$stmt = $conn->prepare("SELECT * FROM users WHERE user_id = ?"); $stmt->bind_param("i", $user_id); $stmt->execute();`

Transactions:
  - Used to ensure data integrity during complex operations.

- Example: Saving a pin to a board and updating the pin count in a single transaction.

Explain How a Virtual Server Was Set Up Locally on Your Device to Mimic a Live Server

I have installed a Local Server Environment called 'MAMP', to create a local server environment with MySQL, and PHP. The server was configured to start Apache and MySQL services. The document root was set to the project directory. PhpMyAdmin was used to create and manage the `trendtrove` database. SQL scripts were executed to create tables and insert initial data. The web application files were placed in the server's document root. The application was accessed via a web browser using `localhost:8888` as the URL.

Explain Techniques Used to Build a Dynamic Web Application

1. Front-End Technologies:
   - HTML/CSS: For structuring and styling the web pages.

   - JavaScript: For interactive elements and AJAX requests. I have used java to document Ready Event, for the notification and chat panels toggle, close panels on outside click, message sending, fetching messages, fetching chat messages, search suggestions and tab navigation.

   - Bootstrap: Have used bootstrap for navbar, for responsive design and pre-built components.

2. Back-End Technologies:
   - PHP: For server-side scripting and handling HTTP requests.
   - MySQL: For database management.

3. MVC Architecture:
   - The application followed the Model-View-Controller (MVC) pattern.
   - Models: Represented by PHP classes interacting with the database.
   - Views: HTML/PHP files that displayed data.
   - Controllers: PHP scripts handling user input and updating models.

4. Session Management:
   - PHP sessions were used to manage user authentication and maintain state across different pages.

5. Security Measures:
   - Input validation and sanitization to prevent XSS attacks.
   - Password hashing for secure storage.
   - Use of HTTPS for secure data transmission.

Test Cases

Test Action	Expected Output	Actual Output	Pass/Fail
Register a new user	User is registered and redirected to login page	User registered and redirected	Pass
Login with correct credentials	User is logged in and redirected to dashboard	User logged in and redirected	Pass
Login with incorrect credentials	Error message displayed	Error message displayed	Pass
Create a new pin	Pin is added to the database and displayed on profile	Pin added and displayed	Pass
Edit user profile	User details updated in the database	User details updated	Pass
Comment on a pin	Comment added to the database and displayed	Comment added and displayed	Pass
Save pin to a board	Pin is associated with the board	Pin associated with the board	Pass
Send a message	Message added to the database and displayed in inbox	Message added and displayed	Pass
Follow a user	User is followed and appears in followers list	User followed and appears in list	Pass
View notifications	Notifications are fetched and displayed	Notifications fetched and displayed	Pass