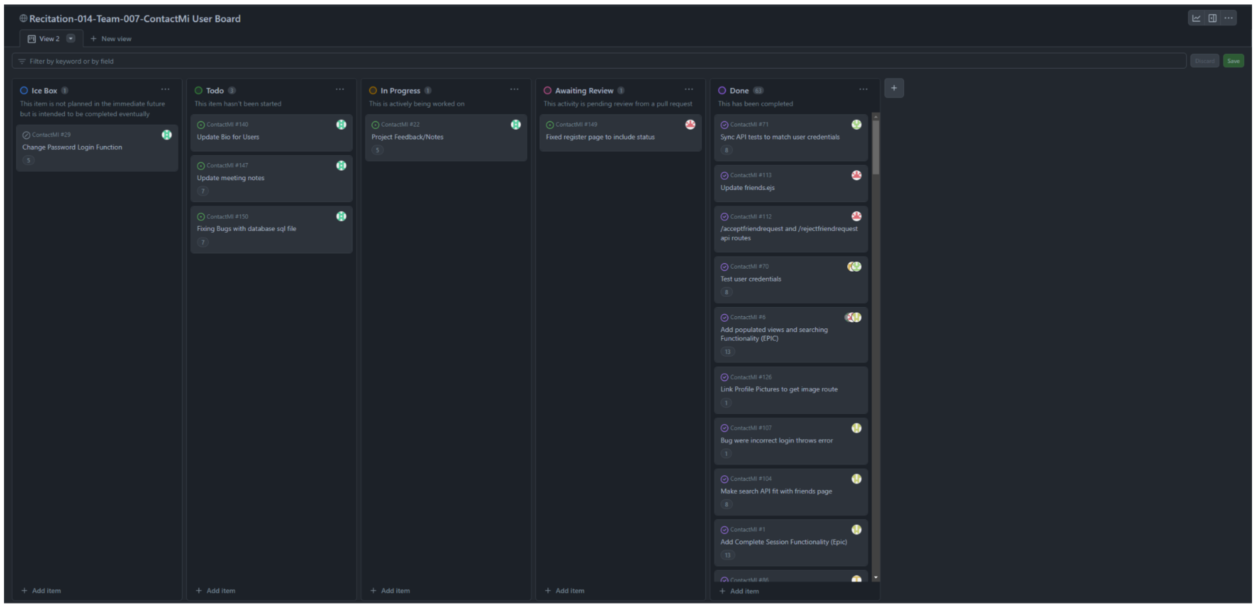
Project Report

Title: **ContactMI**

Who: Vidhaan Singhvi , Samuel Beste , Sansh Goel , Nikhil Bailey , Adan Esparza, Tanmay Meti

Project Description:
ContactMi is a platform resembling Facebook but exclusively focuses on contact information sharing without attention pitfalls like messaging or reels. Users can input their contact details on their profile pages and set statuses for users to see. The home page features a searchable list of users, a favorites display, and detailed user profiles. Login credentials are used for access, and a logout button is available on all pages. This platform diverges from traditional social media, emphasizing streamlined contact sharing over revenue generation through ads.

Project Tracker - GitHub project board:



Recitation-014-Team-007-ContactMi User Board

Ice Box 1 — This item is not planned in the immediate future but is intended to be completed eventually
- ContactMI #29 — Change Password Login Function — 5

Todo 3 — This item hasn't been started
- ContactMI #140 — Update Bio for Users
- ContactMI #147 — Update meeting notes — 7
- ContactMI #150 — Fixing Bugs with database sql file — 7

In Progress 1 — This is actively being worked on
- ContactMI #22 — Project Feedback/Notes — 5

Awaiting Review 1 — This activity is pending review from a pull request
- ContactMI #149 — Fixed register page to include status

Done 63 — This has been completed
- ContactMI #71 — Sync API tests to match user credentials — 8
- ContactMI #113 — Update friends.ejs
- ContactMI #112 — /acceptfriendrequest and /rejectfriendrequest api routes
- ContactMI #70 — Test user credentials — 8
- ContactMI #6 — Add populated views and searching Functionality (EPIC) — 13
- ContactMI #126 — Link Profile Pictures to get image route — 1
- ContactMI #107 — Bug were incorrect login throws error — 1
- ContactMI #104 — Make search API fit with friends page — 8
- ContactMI #1 — Add Complete Session Functionality (Epic) — 13
- ContactMI #86

Link to your Project Tracker:

https://github.com/users/vidhaansinghvi/projects/1/views/2

Video:
https://drive.google.com/file/d/1fwz5TM1nvGQN_m8uU7QK2VE5PCuQT1oB/view?usp=sharing

VCS: https://github.com/vidhaansinghvi/ContactMI

Contributions:

Adan Esparza: My role in the project involved developing key aspects of the user interface, including the registration and login pages. I specialized in handling user-uploaded images for profile pictures. Additionally, I contributed to the backend by creating several PostgreSQL tables and generating test data for these tables.

Samuel Beste: My contributions to the development of ContactMi most involved working on user interface features and background API's. For example, integrating Html pages, Javascript and PostgresQL queries for a smooth user experience. I also assisted with bug fixes and coming up with new features.

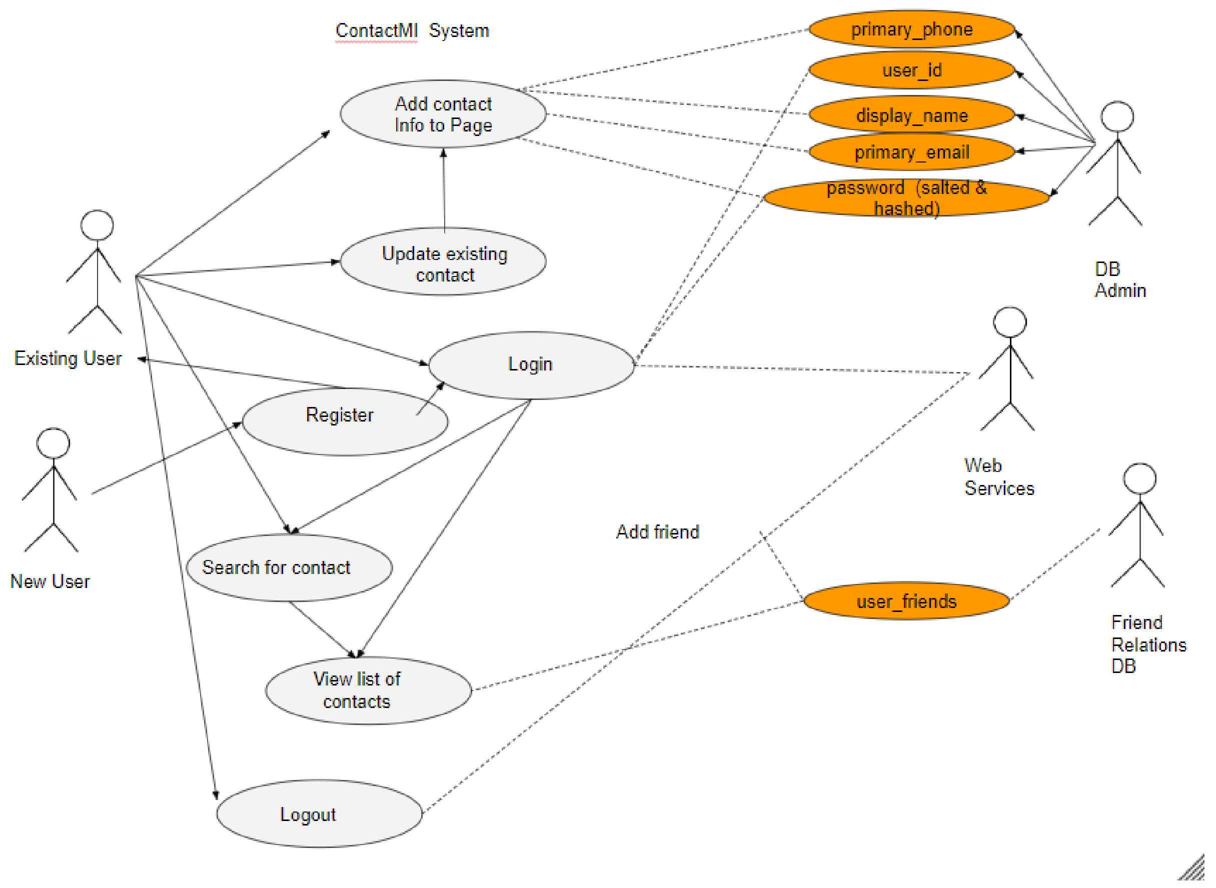
Sansh Goel: In this project, my primary role revolved around ensuring the seamless functionality of our web application. I spearheaded the testing phase, meticulously crafting test cases, executing them, and efficiently pinpointing and resolving bugs within our codebase. Moreover, I took charge of curating and updating the project's GitHub readme, streamlining its accessibility and comprehensibility for users. I also worked on deploying our website on azure host and that was something that was smooth for me without any error.

Nikhil Bailey: In this project, I designed and built the content frame and backdrop, search friends and search users pages, status messages, and the search, friends, and register api routes. I also fixed issues with the development stack, npm, and docker as they aroze. Together with Adan, Sansh, and Sam, I caught and fixed many bugs throughout the application. Additionally, as a very experienced member of the team, I worked with each other developer individually to help resolve their blockers and keep the group moving forward.

Tanmay Meti: In this project, my contributions were mostly focused on developing the home page and the logout page.When I was designing and implementing the homepage and logout page, I was focusing on creating a user-friendly, visually appealing, and intuitive interface for both. In addition to these design tasks, I did the User Acceptance Testing. This involved closely collaborating with end-users to ensure the product met their needs and expectations, as well as identifying and finding any usability issues. My role also included bug fixing, where I successfully identified and resolved some key issues, making the overall functionality and reliability of the platform better.

Vidhaan Singhvi:
In my project contributions, I enhanced user details and fixed profile picture issues, prioritizing a better user experience. I improved interactions, and optimized the search functionality. Additionally, I worked on API routes and a search bar for finding friends. I also organized the users' table for improved database structure. My efforts aimed at making user interactions seamless and enhancing both functionality and appearance in various aspects of the project.

Use Case Diagram: You need to include a use case diagram for your project. You can build on the use case diagram you created in the proposal. If you built a complete use case diagram for the proposal, you can include it as is.



Test results:
Register/ login: the user thought the registration and login went pretty smooth and there were no hiccups or mishaps with the registration. Both users made accounts on the website and logged

in. They thought that a good idea might be to have a prefilled bio so it's not empty for people not willing to answer.

Add Friends feature: The both users added each other and ended up being a seamless process where both the users agreed that it was easy to add each other. One good idea might be is to add a mutual friend feature.

Search Friend Feature: Both the users were able to add each other as friends and had an easy time doing so. When these users added each other, they used the search feature which was easily displayed in front of them and was easy to access.

Deployment: Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.

Link to deployment environment-
This link is to visit our website online through the azure web server.
http://recitation-14-team-07.westus3.cloudapp.azure.com:3000/login

The ContactMi app is deployed using Docker. To run it locally, follow these steps:

Prerequisites: Ensure you have Docker and Git installed on your system.

Running Locally:

- Clone the Repository: Use git clone https://github.com/vidhaansinghvi/ContactMI.git and navigate to the project folder.
- Create Environment Variables: Make a .env file in the root directory and set necessary environment variables for the database and Node.js application.

```
# Database credentials
POSTGRES_USER="postgres"
POSTGRES_PASSWORD="pwd"
POSTGRES_DB="users_db"

# Node variables
SESSION_SECRET="super duper secret!"
```

- Build and Start Docker Containers: Run docker-compose up --build to start PostgreSQL database and Node.js application.

- Accessing the Application: Once containers are running, access the app via http://localhost:3000 in your web browser.

Stopping the Application: To stop and remove the containers, use docker-compose down -v. The -v flag cleans up volumes.

Running Tests: Tests run automatically after executing docker-compose down -v followed by docker-compose up.

This setup allows you to run and access ContactMi on your local machine.