

COMP 4350 Proposal

Project Vision

Caching Feelings is more than just an application; it's a transformative experience for computer science enthusiasts. We want to foster a safe environment where like-minded individuals can connect. The ultimate goal is to facilitate lasting, meaningful relationships that will eventually extend beyond the comforts of the digital realm.

Project Summary

Caching Feelings is an innovative web app designed specifically for computer science enthusiasts, aiming to revolutionize how they connect and interact to enter relationships. The app features a robust and user-friendly registration process where students can create a personalized profile using their email. Profiles include essential details like username, picture, birthday, gender, interests, and academic focuses, such as GitHub repositories or preferred coding languages, enhancing the matchmaking experience. Users will also be provided with the option to remain anonymous or provide their name and photos.

At the heart of **Caching Feelings** is its (Soul) Matching system. It's tailored to help students find matches based on not just proximity and age range but also shared tech interests and academic goals. This ensures that connections are meaningful and based on common passions.

The app excels in communication, offering messaging options, and organized chats for ongoing conversations. This fosters deeper connections among users. The Community Discovery feature allows students to share thoughts, engage with posts, and maintain a safe, respectful community by reporting inappropriate content.

Caching Feelings also offers extensive customization options, including light and dark modes, password change, and anonymous browsing. The integration with GitHub showcases the app's focus on academic achievements and tech skills. Overall, the app aims to make university life for computer science students more connected, less stressful, and intellectually rich, bridging the gap between personal and professional development.

Project Stakeholders

The primary end user of this app will be any individual that has an interest in computer science and is familiar with the terminology and is looking for a partner that shares the same interests. This includes:

- Students

- TAs
- Professors
- CS Alumni
- Working professionals in the tech industry

Another subset of users is those that shy away from traditional dating apps due to social anxiety or because of a more introverted personality. This app will provide a safe space for them to put themselves out there and meet like-minded individuals.

Core features

1. **Account Creation:**
Users can create their accounts and dating profiles
2. **Student (Soul) Matching:**
Users will be able to find matches
3. **Interact with Matches:**
Users will be able to interact with matches
4. **Community Discovery**
Users will be able to view all profiles on the app
5. **Ice Breaker**
Users will be able to send ice breaker message to matches
6. **Non Functional Feature:**
Secure Authentication

Technologies

- MongoDB
- React
- ExpressJS
- NodeJS
- JSON Web Tokens (For secure authentication)
- ThreeJS

User Stories for each feature

Feature 1: Account Creation

Users can create their accounts and dating profiles

- As a user, I want to sign up, so that I can have my own account
- As a user, I want to be able to manage my account
 - Change profile photos
 - Change password
 - Sign out
- As a user, I want to be able to share my interests and preferences
- As a user, I want to be able to add photos to my profile
- As a user, I want to be able to access the account that I created.

Feature 2: Student (Soul) Matching

Users will be able to find matches

- As a user, I want to be able to find people that match my interests and preferences
- As a user, I want to like users, so that I can get to pick the user I like
- As a user, I want to be able to view my matches, so that I can interact with them and know them better

Feature 3: Interact with Matches

Users will be able to interact with matches

- As a user, I want to send and receive messages with my matches, so that we can get to know each other more
- As a user, I want to send read after burn message, so that others can only see it for a certain time
- As a user, I want to be able to view all chats with different students, so that I can continue to chat with them later
- As a user, I want to block matches, so that I will no longer receive messages from them
- As a user, I want to send images in messages to make conversations more fun.

Feature 4: Community Discovery

Users will be able to view all profiles on the app

- As a user, I want to discover people by browsing thoughts in community where everyone can post thoughts, so that I can find people I am interested in
- As a user, I want to post my own thoughts to community, so that everyone can see my thoughts
- As a user, I want to like and dislike thoughts, so that I can express my attitude to that thought
- As a user, I want to delete my own thought, so that others will not see that thought
- As a user, I want the ability to report inappropriate or offensive content in the community.

Feature 5: Ice Breaker

Users will be able to send ice breaker message to matches

* As a user, I want to send a random message to the person I am currently messaging, so that I can start a smoother conversation

Non Functional Feature: Secure Authentication

- User passwords will not be stored in the database and the app will use secure authentication token provided by JSON Web Token

Dev Tasks:

Feature 1: Users can create their accounts and dating profiles

(sprint 2)

- As a user, I want to sign up, so that I can have my own account
 - Home page UI to login or sign up with optional animation
 - Name, DOB, gender identity, email address and password creation UI
 - Confirmation of account setup UI
 - Storage of account identifiers logic
- As a user, I want to be able to manage my account (anonymous mode, UI preferences, modify password and profile, delete)
 - Landing page once user logs in UI
 - UI for settings page
 - Logic to modify database with changes made
- As a user, I want to be able to share my interests and preferences
 - Brainstorm questions we want to ask and tags we want to create
 - UI for user questions and tag selection implemented as an SPA
 - Storage of user selections logic
- As a user, I want to be able to add photos to my profile
 - UI to upload photos
 - Photo storage logic
- As a user, I want to be able to access the account that I created.
 - Login page UI
 - Login information confirmation logic

Feature 2: Users will be able to find matches

(sprint 3)

- As a user, I want to be able to find people that match my interests and preferences
 - Algorithm for matching by interests and preferences
 - UI to select this option
 - UI to show matching in progress
- As a user, I want to match by distance, so that I can find people nearby
 - Algorithm for matching by distance
 - UI to select this option

- UI to show matching in progress
- UI to show matching result
- As a user, I want to be able to view my matches
 - Logic to interact with all matches
 - UI to view matches
 - Logic and UI to favorite matches
- As a user, I want to be able to view my favorite matches
 - Logic to interact with favorite matches
 - UI to view favorite matches

Feature 3: Users will be able to interact with matches (sprint 3)

- As a user, I want to send and receive messages with my matches, so that we can get to know each other more
 - Messages sending and receiving logic
 - UI for users to send and receive message
 - Store messages in local and database
- As a user, I want to send read after burn message, so that others can only see it for a certain time
 - Read after burn logic
 - UI for read after burn
 - Messages should not be stored in database or local
- As a user, I want to be able to view all chats with different students, so that I can continue to chat with them later
 - UI to show list of chats with matches
 - UI to show chat window when user clicked on specific chat
 - Logic to interact with those chat window
- As a user, I want to block matches, so that I will no longer receive messages from them
 - Logic to block matches
 - UI for user to block matches
 - UI to show block status for matches
 - Store blocking information for user in database
- As a user, I want to send images in messages to make conversations more fun.
 - Logic to choose image and send it
 - UI for user to upload image and send it

Feature 4: Users will be able to view all profiles on the app (sprint 4)

- As a user, I want to discover people by browsing thoughts in community where everyone can post thoughts, so that I can find people I am interested in
 - UI to show the community thoughts
 - Retrieve thought data from database
 - Server should be scalable or in round robin approach

- As a user, I want to post my own thoughts to community, so that everyone can see my thoughts
 - UI to allow users to post thoughts
 - UI to show posting result
 - Logic for posting and refreshing in community page
 - Store thoughts in database, each thought with unique id
- As a user, I want to like and dislike thoughts, so that I can express my attitude to that thought
 - Logic to like or dislike thoughts
 - UI for users to like or dislike thoughts
 - Store this information in database
 - On change, modify database information
- As a user, I want to delete my own thought, so that others will not see that thought
 - Logic to delete thoughts
 - UI to allow users to delete their thoughts
 - UI to show deleting result
 - Remove thoughts from database and any information associated with it
- As a user, I want the ability to report inappropriate or offensive content in the community.
 - Logic to report thought or users
 - UI to allow users to report thoughts or users
 - Store report information to database

Feature 5: User will be able to send ice breaker message to matches (sprint 4)

- As a user, I want to send a random message to the person I am currently messaging, so that I can start a smoother conversation
 - UI to send ice breaker message
 - Retrieve ice breaker message from database

Non Functional Feature: Secure Authentication (Sprint 1)

- User passwords will not be stored in the database and the app will use secure authentication token provided by JSON Web Token