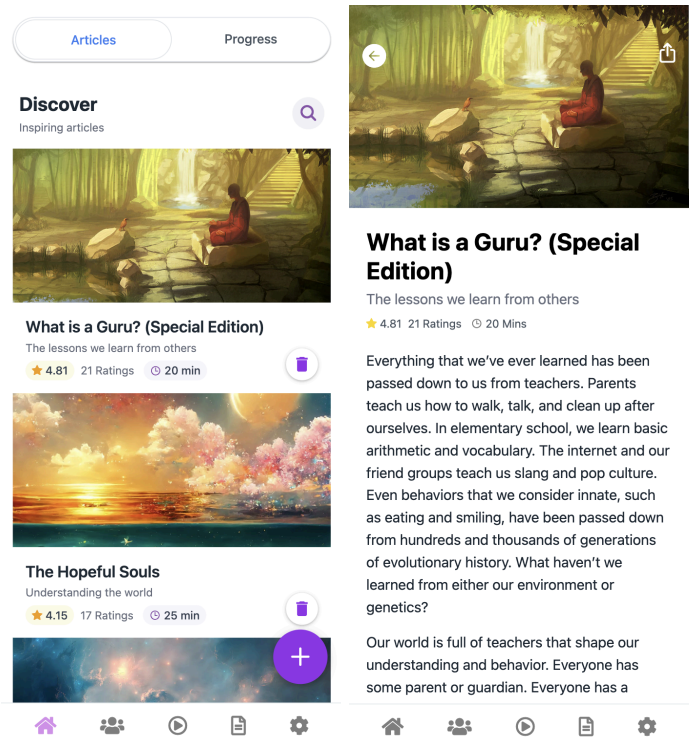
Introduction

Purpose

This document is intended to give an in-depth insight into the project developed by Team 14, SMVS Youth App, for the 2025 Software Engineering capstone symposium. This document outlines the idea and requirements of the project, design, deployment, maintenance, future changes, and challenges experienced.

Project Overview

The SMVS Youth App is an innovative platform that is designed to help foster spiritual growth, mindfulness, and community engagement among young individuals. This is a mobile-focused application that empowers the user by offering a specially curated selection of spiritual content, guided meditation sessions, and personal journaling to help with internal reflection. It ties all of this together with an intuitive user interface that makes the experiences seamless and minimizes distractions so the user can focus on their needs. The app's primary target audience includes youth who seek a deeper connection with the spiritual in this fast-paced, digital-first world. SMVS Youth App V1 is optimized for IOS, but with later versions, there are plans to develop for Android to help support a wider range of users.



About the Client

SMVS (Swaminarayan Mandir Vasna Sanstha) is a Hindu organization with locations worldwide. This organization was chosen as our partner in delivering a new, non-profit product due to the outreach work it has done to connect teens with opportunities to grow their spirituality. Being run by primarily older community leaders, SMVS lacked the tools to modernize and effectively reach younger people in today's society. They asked us to develop the application to help them create custom digital spirituality tools for their organization. On a personal note, many of our team members felt that similar communities were vital in our teenage years. Thus we wanted to help such an organization support its youth population in the digital world.

Project Scope

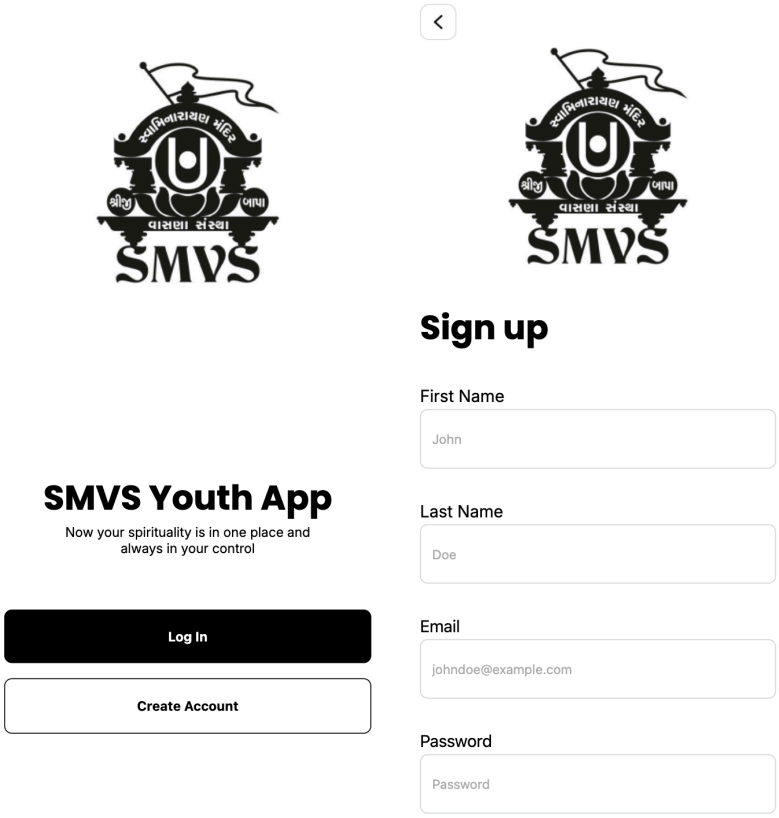
The project's scope is limited to the development and deployment of the main mobile application. Beyond initial deployment and handoff, the client is expected to maintain the app themselves without support from the FYDP team. There is also no expectation of future feature development, nor are there any warranties provided with the software. Users of the software will agree to the terms and conditions of the SMVS organization. The FYDP team is not responsible for paying for, or managing, any cloud instances associated with the application. Although best efforts have been made to follow industry-standard security practices, the FYDP team is not responsible for the security integrity of the software.

Core Functionalities

The core functionalities of the Youth App can be explored in just a few steps. To start, download the app from the IOS App Store. Launch it and follow these steps:

1. Login/Sign Up Once the app is open, you can log in with an existing account or sign up using a valid email address.
2. Set Preferences Navigate to the user preferences tab, where you can customize the type of spiritual content categories you want to focus on, such as meditation, mindfulness, scriptures, and motivation.
3. Access Guided Meditation Join a live meditation session from the Meditation tab or look in Videos for pre-recorded sessions for a self-paced meditation experience.

4. Journal Reflection Access Journal Reflection through the Journals tab where you can add entries about your spiritual reflection or daily gratitude.
5. Explore Spiritual Content Browse your personalized content via the Explore tab and see all your recommendations.



System Requirements

Software Requirements

- Server: Python 3.9+, FastAPI or Django framework
- Database: PostgreSQL 14+
- Frontend: React Native 0.70+, TypeScript 4.5+, Tailwind CSS 3.0+
- Authentication: Google Firebase Authentication
- Storage: Google Firebase Storage for media files
- State management: Redux
- API format: RESTful or GraphQL endpoints

Development Requirements

- Node.js 16+ for frontend development
- Python virtual environment management (venv)
- Git version control
- CI/CD pipeline integration (GitHub Actions or similar)
- ESLint and Prettier for code formatting
- TypeScript strict mode enabled
- Jest for frontend testing
- Pytest for backend testing

Deployment Requirements

- Docker containerization for consistent environments
- Environment-specific configuration management
- Database migration strategy
- Automated deployment pipeline
- Monitoring and logging tools integration

Performance Requirements

- API response time < 200ms for standard operations
- Application load time < 3 seconds on 4G connections
- Database query optimization for frequently accessed data
- Image and media optimization for mobile devices
- Firebase Storage CDN utilization for faster content delivery

Network Requirements

- Bandwidth: Minimum 10 Mbps for video streaming
- Latency: <100ms for real-time meditation sessions
- SSL/TLS encryption for all data transmission
- CDN integration for content delivery optimization

Security Requirements

- End-to-end encryption for user messages
- Data encryption at rest for personal journal entries
- Role-based access control (RBAC) for admin functions
- Regular security audits and penetration testing (client responsibility after handoff)

When spiritual content is uploaded or updated by an admin, the following will occur:

Database Tables Updated:

- Content Table: Stores the spiritual content, including title, description, locations, tags, media URL, timestamp of the update, and the id of the content.
- Preferences Table: This table maps the user IDs to their selected content preferences. These content preferences are a preset enum list. This ensures new content can be correctly displayed for each of the users.
- Audit Log Table: This table stores all changes made. Such as the admin's user ID, timestamps, and the change type (add/edit/delete).
- Indexes: Compound indexes will be used based on categories or tags for efficient retrial of content.

Sample Workflow Summary

1. The admin adds or modifies spiritual content.
2. The back-end will validate the input by the admin and update the tables in the database respectively.
3. The updates will be cached in Redis, and then a push notification will be triggered through our WebSocket integration.
4. The front end will refresh dynamically to display the newly updated content.
5. The user will be able to see the newly updated content.

Limitations

- The guided meditation sessions will require a stable internet connection.

- The app supports the English language for version one.
- The user's content recommendations depend on the accuracy of the user's initial preferences; limited customization options may impact the relevance of the displayed content.

Non-Functional Requirements

Performance Metrics

Performance shall be measured through system response times and user engagement and enjoyment.

Various metrics about system performance gathered included API response times, page loading speeds, database query speeds, network delays, 3rd-party API response times, and more. Combined with user testing, these metrics were used to verify that the system performed well under various load levels. Further system testing used backed (pytest) and frontend (Jest) testing frameworks. The use of these testing frameworks validated the correctness of key functionalities, as well as their performance under some levels of load.

Performance under real-world scenarios was measured primarily during user testing. In trials, starting first with community leaders and then with community participants, we were able to test and validate assumptions made during the development process. Initial tests with the community leaders led to lots of iteration on the product and a growing understanding of what exactly the client wanted to be delivered. Once the product was in a satisfying state for the client, we could onboard a small user group (10 youth) to test the look and functionality of the app. Initial reactions were mostly positive, with some suggestions on adding features (ie. AI Avatar Generation) to further enhance user experience across the age groups tested. Towards the end of the development process, user-related performance was deemed positive, and there was a lot of feedback about the usefulness and usability of the app for the specific use case.

Scalability Considerations

The initial app was developed for an organization with about 150 youth in the community. For such a small community, scalability is not a significant concern since the free tier of many cloud services could handle such loads. In our performance testing

with simulated users, we saw no significant slowdowns for this number of users. However, the SMVS organization does have chapters across the world. If the single instance were expanded to all members of the SMVS community worldwide, slowdowns are very likely. There was no sharding of the DB, nor was there any caching of data or locality-based loading. Such features would ensure a seamless experience with a larger amount of clients.

Instead of upgrading the existing existence of the app, our expansion plan is to launch new instances of the application - an instance per organization we are working with. For example, since the app developed effectively as a form of contained and moderated social media, we considered partnering with schools to deploy the app as an introduction to safe social media usage. In such a use case, we would redeploy the app as a new instance with custom branding for the school rather than a copy of the same instance.

Accessibility features

Although accessibility wasn't a primary goal for the first iteration of the app, we did make sure to keep in mind some accessibility concerns during the design and development process. Accessibility features developed in this first iteration include:

- **Color Contrast:** Design text and banners with contrasting colors for easier visibility
- **Responsive Design:** Adapts to different screen sizes and orientations
- **Meditation Audio**: Transcripts provided for all guided meditation sessions
- **Alternative Text**: Provide admins with the ability to create alternative text/captions for all images uploaded onto the system

In future iterations of the application, accessibility will be a forefront concern. Particular features that would have significant positive impact would include:

- **Reduced Motion Option**: Ability to disable animations for users
- **Screen Reader Compatibility**: All UI elements are properly labeled with aria attributes for screen reader compatibility
- **Keyboard Navigation**: Complete functionality available through keyboard controls

Privacy and data protection

User privacy and data protection were the utmost priorities when developing this application. From the design phase, we wanted to limit the amount of Personally Identifiable Information (PII) gathered from the client. Besides collecting basic information such as name, location, and email address, our application does not collect much PII. Any collected information is securely stored in Postgres, with read access controlled through user permissions. General users of the platform can only access their data, along with any publicly available data in the app (i.e., feed posts from other users and article contents from admins). Admins have much broader access permissions, including being able to query for user-generated content. By design, our application doesn't include any private peer-to-peer communication capabilities to reduce harmful behavior.

User passwords are never stored on the system but are handled by Google Firebase's battle-tested auth system. For data stored in Postgres, we made all attempts to ensure the data was secure and our APIs could not be breached using common infiltration methods. We use Authentication Bearer tokens to validate user access and revoke tokens (requiring refresh) at regular intervals. Our app also does not ask for user permissions from the user's phone, besides notification permissions, to reduce our intake of the user's data.

Design

Layout

This Youth App is designed to have a clean and intuitive interface to help minimize distractions and make user interactions straightforward and efficient. The app's main layout is divided into distinct sections to support user activities and any separate admin tasks, each accessible through a navigation tab on the bottom. Having this present on all screens allows for a seamless transition between functionalities for the user.

Navigation

The navigation bar provides efficient and quick access to our primary app features.

- Home: Displays recent messages, updates, and any notifications regarding upcoming events. The user can edit what is displayed on their home screen and prioritize some content over another.

- Journal: A personal space where the user can document anything their heart desires by allowing the user to either record or write their journal and use our search feature to view their past journals to see their spiritual journey.
- Meditation: Enables the user to join either a live guided session or access prerecorded meditation content.
- Settings: Personal setting option where the user is able to modify their personal information or their spiritual content preferences, such as selecting preferred categories of content or their notification preferences.

User Interface Elements

The Youth App incorporates various interactive elements designed to enhance user experience:

- Content Library: A list of featured content that includes mediation, news, events, and other spiritual topics. After content is selected, it will redirect the user to the selected content such as an article, video, or audio.
- Community Forum: This is a community section where the user is able to view posts, participate in general discussions, and connect with other users. Forums are tagged with key labels so that the user is able to select what type of forums they want to view/participate in.
- Explore Menu: The user can use this menu to explore new videos, events, mediation, or public forums. This will also include the filters identified above such as type of content, location, and labels regarding the content so the user is able to be granular in what they want to explore.
- Form for Content Upload (Admin): Admins have access to our content upload section where they are able to add new content by specifying the content name, location, specific tags, and a description of what the content is. This allows for easy content management and updating content through the app

User Use Cases

Modify Spiritual Content Preference

This use case is what allows the user to personalize the spiritual content that is displayed. The user can customize based on various categories such as mediation, scripture readings, type of media (video, audio, articles), spirituality, self-development, personal growth, mindfulness, etc. These preferences will help tailor the content

recommendations which will allow the user to focus on areas of interest for a more personalized feel.

Steps:

1. Open the App: Launch the SMVS Youth App and ensure the user is logged in for the correct account they want to adjust the preferences for.
2. Navigate to Settings: From the navigation bar at the bottom, tap on the Settings icon to view the settings menu.
3. Select Spiritual Preferences: Within this menu, locate and select the Spiritual Preferences tab. This will open a submenu with options that are related to customizing the users' spiritual content.
4. View Current Preferences: The app will display the users' current preferences, which will show categories like Meditation, Scripture, Mindfullness, and SelfDevelopment. Each category will have a toggle switch or checkbox to indicate whether it's currently selected.
5. Adjust Preferences:
    a. Toggle Categories: Turn each category on or off based on your interest by tapping the toggle switch next to it.
    b. Set Frequency: For each active category that is selected, the user can adjust how frequently they want to receive updates (e.g., Daily, Weekly, Monthly). With a dropdown menu appearing, the user can select their desired frequency.
    c. Subcategories (Optional): Some categories, such as Meditation, may have assigned additional subcategories (e.g., Guided Meditation, Mindfulness, Breathing Exercises). Follow the steps above to edit these sub-category preferences.
6. Save Changes: Once the user is finished with setting their preferences, scroll down and tap on Save Changes button on the bottom of the screen. The user should expect to see a new window that has the text "Preferences updated successfully" which will signify that the user's preferences are saved.
7. Exit Settings: To return to the home screen, tap the back arrow or home icon in the bottom navigation bar.

Add Spiritual Reflections or Entries in Journals

This use case enables the user to add their spiritual reflection or daily thoughts into their journals within the SMVS Youth App. This feature is designed to encourage mindfulness and self-reflection by providing resources to where to document the thoughts, gratitude, or any spiritual insight. Tags are used to allow easy searching for the user's past journals and see their growth for a specific topic.

Steps:

1. Navigate to the Journal Tab: Open the app and using the navigation bar, tap on the Journal tab. This will open your personal journal dashboard and display any existing journals.
2. Start a New Entry: Tap the + (Add) button located at the bottom right corner of the journal dashboard. This action will result in a new entry screen with a text area for the user's input for their reflection.
3. Enter Reflection: After this new screen is opened, the user will be able to type their spiritual reflection, thoughts, or daily gratitude in the text area provided. This is similar to a mark down text editor that supports formatting options such as bullet points or numbered lists for a more structured journal entry.
4. Add Tags (Optional):
    a. Below the text area, locate the Tags field.
    b. Tap the field and enter relevant tags such as Gratitude, Meditation, or Daily Reflection.
    c. Separate multiple tags with commas or select from a list of suggested tags based on previous entries.
5. Attach Media (Optional): If desired, add an image or audio recording to enhance the journal entry:
    a. Tap the Attachment icon below the text area.
    b. Choose Image to upload a photo from your device.
    c. Choose Audio to record a voice note directly within the app.

6. Save the Entry: Once your reflection and any optional additions are complete, tap the Save button at the bottom of the screen. A confirmation message, ”Entry saved successfully!”, will briefly appear to indicate that the entry has been stored. Once the user is satisfied with their journal entry, tap the Save button at the bottom of the journal screen. There will be a confirmation message, ”Entry saved successfully!” as a popup.
7. Review and Edit: Return to the journal dashboard to see your newly added entry. You can tap on the entry to view or edit it at any time.

Join Guided Meditation Sessions Online

The guided meditation feature allows users to participate in live, online meditation sessions that are led by experienced instructors. This feature helps foster mindfulness, relaxation, and hopefully a spiritual connection through a shared, real-time mediation experience. The user can browse upcoming sessions, view the session details, and also join these sessions through the app.

Steps:

1. Navigate to the Meditation Tab: Open the app and use the navigation bar to select the Meditation tab. This will then display a list of available guided meditation sessions. Whether that’d be live, pre-recorded, or any upcoming sessions.
2. Browse Available Sessions:
    a. The user can scroll through the list of meditation sessions to view details about each one. This includes details such as the session topic, duration, instructor, start time, ect.
    b. The user can use filters, if available, to narrow down sessions by type, duration, or instructor.
3. Select a Session to Join: Once the user finds a session of interest, the user can tap on it to open the session details page. Here, you can see information like the session’s theme, level of difficulty, or any requirements for the session (e.g., quiet environment, comfortable seating).

4. Tap Join Now: • For live sessions that are currently available, a Join Now button will be available in front of the session box.
    a. Tap Join Now to enter the session.There might be a brief loading screen before the user is joined into the session.
5. End of Session: Depending on the sessions host, a thank you message for participating might pop up. The user will have the option to rate the session or provide feedback. The user can navigate to other parts of the app now using the navigation bar at the bottom.

Further Features:

- Autosave Feature: This app will autosave the user's progress when they are writing in their journal text area. This save feature is automatically running in intervals of 30 seconds to help prevent data loss during entry creation.
- Privacy Settings: All journal entries are private and accessible only to the user. The app does not share or sync entries externally.
- Exporting Entries: Users can choose to export their journal entries as a PDF document by selecting the Export option within the journal dashboard.
- Daily Reminder: The user can enable daily notifications to remind them to add reflections by visiting the Settings menu under Reminders
- Session Reminders: The user can set reminders for any upcoming sessions they are interested in by tapping the Set Reminder button on the session's details page. This will send a notification to the user, depending on their notification preferences, 5 minutes before the session begins.
- Follow-Up Content: After a live session and the user's feedback, the content on the user's explore page may be updated depending on the feedback given by the user.
- Session Recordings: Some sessions may be recorded and made available for replay. Users can access recorded sessions from the Meditation tab under a "Past Sessions" section.
- Privacy Considerations: The user's participation in sessions is private and not visible to other participants.

Admin Use Cases

Manage Spiritual Content

Steps

1. Log in to the app with administrative credentials: The admin must log in using their unique credentials to access the admin dashboard and use admin permissions. If login credentials are not valid, an error will be thrown and the admin will be requested to attempt to login again.
2. Navigate to the Content Management section: Once the admin is logged in, locate and tap the Content Management tab in the navigation bar or through the admin dashboard. This is the section that is designed to manage spiritual content for the users on the SMVS Youth App.
3. Add new content:
    a. In the content management screen, tap the Add Content button.
    b. Fill out the form fields, including: – Content Name: The name of the spiritual content. – Description: A brief summary or overview of the content. – Tags: Keywords to help categorize the content (e.g., ”Meditation,” ”Mindfulness”). – Location: Location of where you want this content to be displayed (e.g., Toronto, California, Texas). – Video or Document upload: Click on either the video or document icon right above the Upload button to upload your video/mp3 or any article/document required.
    c. Once all details have been entered and verified, tap the Upload button to save and publish the content.
    d. A confirmation dialog will appear, confirming that the content has been successfully added.
4. Edit existing content:
    a. Select the content the admin wish’s to edit from the list of available content in the Content Management section.
    b. Tap the Edit button, top right, which will open the content in an editable form. Will look similar to what is displayed above for uploading content.

c. Make the necessary changes, such as updating the title, description, tags, or media file.

d. Tap the Save Changes button for changes to persist.

e. A success notification will confirm that the edits have been saved. This update should shortly be propagated to the users.

5. Delete content:

    a. Select the content the admin wishes to delete from the list of available content in the Content Management section.

    b. Tap the Delete button.

    c. A confirmation prompt will appear, asking the admin to confirm the deletion.

    d. Tap Yes, Delete to delete the content.

    e. A success message will appear, confirming the content has been deleted. The content will no longer be displayed to the user.

Potential Errors and Resolutions

- Error: Failed to upload content
    - Check that the type fo content the admin is uploading aligns with the app's capabilities in format and size.
    - Retry the content upload with a stable internet connection and after verifying the information above.
- Deleted content isn't immediately deleted.
    - Sometimes, if a user is accessing the content that is deleted by an admin, it might still be accessible to the user since it's most likely already loaded on their end and stored on their local cache.
    - If admin wants to ensure that no one see's deleted content and have that propagate immediately, the SMVS Youth App can send a refresh request to the user such that it will need to get the content again and when it tries to, since it's deleted, it wont be displayed.

Testing and Quality Assurance

Unit Testing

Using the Pytest library, we created unit tests for individual components and functions in the backend. These tests involved validating API endpoints and database interactions, as well as making sure that our class design and helper functions worked as expected.

Integration Testing

We also added integration tests to check the interactions between different parts of the system. For example, we tested the communication between the React Native frontend and the backend, as well as all interactions between our code and the Firebase Authentication and Storage APIs.

System/End-to-end testing

We also manually tested processes that simulated end-to-end user workflows. For example, we individually went through the full sign-on workflow, setting preferences, journaling, meditation, and upload features. We tested all these workflows locally and after deployment on iOS, Android, and web on both simulators and actual devices.

Performance Testing

As detailed in the non-functional requirements, performance was evaluated primarily through metric collection and real-world validation. Some of the metrics we collected were related to API response time, application load time, and database query efficiency.

User Acceptance Testing

The first user group that helped us test the app was the SMVS youth leaders. This phase involved a lot of iteration, as we had to clarify requirements and refine the app's core functionality and design based on their feedback. We also had a target user group of 10 youth participants, who tested the application when it was nearly complete and identified areas for improvement and proposed their own desired features.

Accessibility Testing

Finally, we did manual checks as we developed our frontend to make sure that we were meeting our initial accessibility requirements.

Deployment Information

To deploy our app, there were a few steps we had to follow. First, we containerized the entire backend and database using Docker. Since we use Google Firebase for user authentication and media file storage, we didn't have to make any changes to support the features that interacted with those APIs. Next, to handle environment variables such as API keys and database connection strings, we managed these using a configuration file that is inputted during the deployment process to avoid any sensitive data being accessible. Finally, we went through the setup process on Heroku to deploy the dockerized backend so it can be accessible from any frontend.

To submit it to the app store we had to build an IOS application archive and get approval to be released onto the app store. The SMVS organization is responsible for managing the costs and releases after the initial handoff.

Maintenance and Support

After the initial deployment and handoff, the SMVS youth organization is responsible for maintaining and supporting new features for the app. However, the team will be available to help with any issues. Specifically, if there are any bugs that are discovered after deployment, we will be responsible for addressing them. As part of the handoff, we will onboard the members of the SMVS youth organization responsible for maintenance on the structure of the codebase, how each feature was implemented, and how to conduct basic tasks such as performing database migrations and monitoring the app performance. The rest of the tasks, including addressing user feedback and requests as well as app store and cloud management will be handled by members of the SMVS team.

Future Development

Some potential developments in the future include:

- ☐ Developing a native Android version of the app to support a wider audience
- ☐ Improving the accessibility of the app including screen reader compatibility and better keyboard navigation
- ☐ Scalability improvements such as adding a CDN for content distribution, load balancers, or partitioning the database

- [ ] New features such as more gamification and expanded community features like live meditation sessions or mood tracking
- [ ] Support for multiple languages
- [ ] Advanced personalization using better recommendation algorithms