DyslexiaAI
Development Plan

Project Members:

Saba Begum

Mahbubur Khan

Aaron Perel

Ishmael Gonzales

1. Project Overview

This project aims to create a web application designed to support individuals with dyslexia, called the DyslexiaAI. By visiting this web application, users will benefit from features specifically tailored to make content easier to read, understand, and navigate. Our website will provide access to books and stories, integrating them into a dyslexia-friendly format.

When a user accesses the Dyslexia-Friendly Web Application, they will be prompted to log in. First-time users will need to sign up. Once logged in, users will have access to a range of services designed to enhance their reading experience. These services include the option to select from a provided list of books or choose other e-books. The books will then be presented in a dyslexia-friendly format.

Key features of the dyslexia-friendly presentation include:

- Font Size: Larger font sizes (minimum 12-14 points).
- Font Choices: Options like Arial, Verdana, or Open Dyslexic.
- Line Spacing: 1.5 to 2 times the font size.
- Paragraph Spacing: Increased space between paragraphs to break up text.
- Contrast: Extreme contrast between text and background.
- Line Length: Keeping lines between 60-70 characters to avoid overwhelming the reader.
- Emphasis: Using bold for emphasis instead of italics or underlining.
- Background Color: Off-white, cream, or pastel backgrounds, as these are often easier on the eyes compared to pure white.
- Deaf Mode:
- Blind Mode:
- Book Recommendation:

This project will provide a valuable resource for individuals with dyslexia, making reading more accessible and enjoyable.

2. Project Purpose, Scope, Objective

Approximately 20% of students in the USA have dyslexia, a condition that makes reading and writing challenging. Despite these difficulties, reading offers numerous benefits, such as improving memory, enhancing concentration, and reducing stress. This project's purpose is to develop a dyslexia-friendly web application that allows users to easily access and enjoy these benefits through its user-friendly features.

Front-End Implementation:

1. Login Screen:
    - Username and password fields with large input areas and clear labels.
    - Prominent login button with extreme contrast color.
    - "Forgot Password" link located under the login button for easy access.
2. Home Page:
    - Navigation bar with links to Home, My Books, Settings, and Logout.
    - Prominently placed search bar for finding e-books.
    - Book categories section listing different books.
3. Reading Page:
    - Adjustable text with options to change font size, font type, and background color.
    - Text-to-speech functionality with buttons to activate it.
    - Navigation buttons for the next/previous page.
4. Settings Page:
    - Profile information section with options to edit profile details.
    - Reading preferences section to save preferred font size, type, and background color.
    - Accessibility options to enable/disable text-to-speech.

Back-End Implementation:

1. User Authentication:
    - Secure database to store user credentials.
2. E-Book Management:
    - Database to store book information, categories, and user selections.
3. User Preferences:
    - Database to store user-specific settings and reading preferences.
4. API Endpoints:
    - Authentication endpoints for login, signup, and password recovery.
    - Book data endpoints to fetch book categories and details.
    - User preference endpoints to save and retrieve user preferences.

Objective:

This project's objective is to help individuals with dyslexia by making it easier for them to read books.

3. Team Organization

The development team includes Saba Begum, Ishmael Gonzales, Mahbubur Khan, Aaron Perel. Each will perform individual tasks based on the assigned roles and keep the project up to date and the source code well maintained. Each person will have their own branch to avoid code conflicts and minimize pitfalls. During team meetings, code changes will be discussed to ensure

no bugs or issues are present in the merger of all branches to main for the updated development build or production build.

3.1 Sub Teams:

Team Lead:
Aaron Perel

Frontend Team:
Aaron Perel, Mahbubur Khan

Backend Team:

Ishmael Gonzales, Saba Begum

Database Lead:

Saba Begum

Documentation Lead:

Ishmael Gonzalez

QA Lead:

Mahbubur Khan

Presentation Lead:

Aaron Perel

3.2 Roles and Responsibilities:

Aaron Perel:

- Managing Code Base via GitHub using a remote repository, in specific authorizing Pull and Push requests, clearing issues, and approving Merge requests.
- Organizing meetings and keeping a common consensus within the group.
- Frontend development of components, systems, and UI Design.
- Assisting with the Backend Development.

Mahbubur Khan:

- Frontend Development using React, Tailwind CSS and more to create a dynamic WebApp / website to display tools.
- Styles and UI design for websites individual pages and tools.
- Organization of personal branch on GitHub.

Ishmael Gonzales:

- Backend Development, using JavaScript via Node.js and the Firebase JS SDK, of authentication requests, data / user storage, and front-end requests from the web server.
- Organization of personal branch on GitHub as well as pull requests and issue logging.
- Assisting with UI Design and UX design via styles and Figma.

Saba Begum:

- Backend Development of API requests to AI (Artificial Intelligence) model that will handle image generation, text-generation, and requests made to any other sources outside of Firebase.
- Organization of personal branch on GitHub as well as pull requests and issues.
- Assisting with Frontend development of component architecture and components

4. Problem Resolution Policies:

4.1 Raising Issues:

Anytime a major issue arises it will be brought up via our developer group chat, this issue will then be logged in some way shape in a google document to keep track of all current issues. If the issue is code related, then an issue will be created in the remote repository on GitHub as well. Based on what the issue pertains too, whether it be personal, code, or document related it must be discussed with the team.

4.2 Discussing Issues:

Issue discussion will be done during team meetings. These discussions will be focused on the issues contained within the google doc and all those on the GitHub repository. During the discussion issues will be delegated to team members (if code based) based on their sub team and their expertise.

4.2.1 Code Issues:

Code issues will first be introduced by how they were created and or found by the team member(s) that found them. After this an analysis of the root cause of said issue will be discussed by the team and evaluated. After finding the root cause and its impact on the project the issue will then be marked with a rating of critical, high, or medium. The issue is then assigned to a sub team and / or a team member based on what they are currently working on.

4.2.2 Team Issues:

Team issues will be discussed within the group chat or during team meetings and necessary measures will be taken to accommodate. If a member cannot complete the work, certain work will be re-delegated from team members to better accommodate them with the issue. If the member causes other issues like not attending the meeting, etc., we will get the professor involved.

4.3 Closing, and Resolving issues:

Resolving or delegating issues will be the first step of each meeting to guarantee everyone is accommodated and all known code issues are acknowledged or flagged.

5. Project Plan:

Our team is committed to presenting each project deliverable on the assigned due dates. These presentations will be conducted in-person in front of the class or via Zoom, ensuring accessibility for all team members. The meeting schedule is set for either Tuesdays or Thursdays during the class hours from 6:30 pm to 8:00 pm, accommodating everyone's availability and ensuring effective communication and collaboration.

5.1 Project Development Plan: Thursday, 5/23/24

- Give an overview of the project.
- Present project's purpose, scope, and objectives.
- Outline team organization roles and responsibilities of the team members.
- Present the team management plan and the technology to be used for our project.
- Development Plan presentation.

5.2 Software Requirements Specification (SRS): Thursday, 5/30/24

- Gathering all the information for our project from the team members.
- Understand both functional and non-functional UI particulars.
- Prioritize the features, functions, and non-functional UI.
- Software Requirement presentation

5.3 First Prototype: Thursday, 06/06/24

- The team should produce a specific UI design and have the front-end ready.
- Start testing and prepare to create the test plan.
- Present the prototype to the class.

5.4 Software Design Specification: Thursday, 06/18/24

- Requirement Analysis: Understand and document the project's needs.
- System Architecture Design: Define high-level system components and interactions.
- Detailed Design: Specific detailed functionality and component relationships.
- Interface design: Design user interface and software communication methods.
- Security design to protect against threats.
- Performance design to optimize software for efficiency and scalability.
- Documentation: Create comprehensive design documentation
- Review and Validation: Ensure completeness and alignment with requirements.

5.5 Second prototype: Thursday, 06/25/24

- Have most of the project ready.
- Present the second prototype.

5.6 Test Plan: Thursday, 07/02/24

- Identify Test Objective: Define what needs to be tested.
- Defining Scope: Determine what features and functions will be tested.
- Select testing Techniques: Choose appropriate methods like unit testing or user acceptance testing.
- Create test cases: Develop detailed steps and expected results for testing.
- Identify test data: Gather essential data for effective testing, including valid and invalid inputs.
- Schedule testing: Planning timing and sequence of testing
- Review and approve the test plan.
- Present test plan

6. Configuration management Plan:

6.1 Code Storage:

All code created by the team will be stored on a remote repository on GitHub. Each member will have their own branch to maintain control over the most updated and working development build.

6.2 Push, Pull, Merge:

Push, Pull or Merge requests will be peer-reviewed by the other sub-team members and the team leader. Once the code is reviewed, only then, a push or merge request will be completed.

7. Technologies:

Throughout this project's development process, we will use the following technologies and tools listed below for the back end and front-end development of the website.

7.1 Front-End Technologies:

Frontend Development

React Js framework will be utilized throughout the development process of the front-end for the completion of the website. We chose to use React Js because of its simplicity, JavaScript UI library, and reuseable component base architecture.

Libraries

The functionality of the website depends on the implementation of multiple libraries such as Tailwind, and ResponsiveVoice will be used and many more for the website to work effectively.

7.2 Back-end technologies:

Database

Utilization of Firebase platform for managing the backend infrastructure of the project such as the authentication process and for data storage and management. We are using Firebase as our

cloud service because of its specialization of server-side language, easy-to-use authentication system, and its third-party provider sign-up options.

Runtime Environment

Node.js, we will utilize Node.js by using it for Api requests, and web server creation and debug logging.

Api testing will be taken care of via Postman.

Unit tests will be carried out via built in react unit test tools as well as the React Testing Library allowing us to test DOM benchmarks.