

Project Report

EduAid

Group Members	CMS	Roles
Muhammad Talal ...	365776	Frontend Developer
Muhammad Uzair ...	368187	Backend Developer
Umar Tariq	368170	UI designer

Submitted to: Sir Fahad Ahmed Satti

Date of Submission: 17-05-2023

Table of Contents

Introduction.....	2
Problem Statement.....	4
Requirement Elicitation.....	5
Implementation.....	7
Backend.....	7
Database Selection:.....	7
Database Schema:.....	7
Database Connectivity:.....	7
API Endpoints and Routes:.....	7
User Registration:.....	8
User Authentication:.....	8
Dashboard:.....	8
Frontend.....	9
Data Flow Diagram.....	9
Further Improvements.....	10
Hosting:.....	10
More Activities for Treatment:.....	10
Audio-Based Learning:.....	10
Interactive Exercises for ADHD:.....	10
Guided Meditation and Relaxation Techniques:.....	10
Peer Support and Community Features:.....	10
Installation and Usage.....	11
Prerequisite:.....	11
Conclusion.....	14



Introduction

The purpose of this project report is to present the development and functionality of a web application designed to cater to individuals with visual difficulties, including Dyslexia, ADHD, and Visual Agnosia. The web application was built using the **MERN** (MongoDB, Express.js, React.js, Node.js) stack, providing a comprehensive and user-friendly platform to assist individuals in enhancing their cognitive abilities and managing their learning plans.


The primary objective of this web application is to offer a supportive environment for users with visual difficulties, empowering them to overcome challenges and improve their overall learning experience. By incorporating various features such as **diagnostic tests, memory games, word scramble games, and personalized learning plans**, the web application aims to address the specific needs of individuals with different visual difficulties.

The web application consists of distinct components, including a Login and Signup page accessible through the Landing Page. User registration and authentication are facilitated by securely storing user data in a MongoDB database. Once registered, users gain access to a range of features designed to evaluate and enhance their cognitive abilities.

One significant aspect of the web application is its diagnostic test feature, which allows users to identify their specific visual difficulty. By generating a personalized ticket based on the test results, users can gain a better understanding of their condition and tailor their learning experience accordingly. Furthermore, the inclusion of memory games and word scramble games provides engaging activities aimed at improving cognitive abilities.

To track and monitor user progress, the web application incorporates a scoring mechanism. This feature enables users to keep track of their performance across different levels, providing a sense of accomplishment and motivation for further improvement. Additionally, users have the ability to create and manage their learning plans, allowing them to focus on specific difficulties and set personalized goals.

Throughout this project report, we will delve into the various components, features, and technologies used in the development of the web application. We will discuss the design considerations, implementation details, and the overall impact and potential benefits of the application for individuals with visual difficulties. The subsequent sections will provide a detailed overview of each component and feature, highlighting their functionalities and how they contribute to the user experience.



By the end of this report, it is expected that readers will have a comprehensive understanding of the web application's purpose, functionality, and the potential it holds in supporting individuals with visual difficulties.



Problem Statement

In Pakistan, there are around 220 million people, and studies show that about **10-15%** of school-aged children face learning difficulties like dyslexia, visual processing issues, and ADHD. These difficulties make it hard for them to learn and do well in school, which can affect their future success. Unfortunately, many people in Pakistan don't know much about these learning difficulties, so children often don't get the help they need.

Even when a child is diagnosed with a learning difficulty, it can be tough to find the right support. Regular classrooms usually don't have the right tools and methods to help these children, which makes them feel frustrated and anxious. And many families can't afford special services or resources, so their children miss out on the help they require.

All these challenges make it difficult for children with learning difficulties to succeed in school. Without the right support and accommodations, they struggle to learn, and it affects their confidence and overall well-being.

To address these problems, we have developed a web app specifically for people with visual difficulties like dyslexia, ADHD, and visual agnosia. The web app includes diagnostic tests to identify specific difficulties, games to improve cognitive abilities, personalized learning plans, and a scoring system to track progress. By creating this web app, I hope to provide a solution that is easy to use and helps individuals with visual difficulties overcome their challenges and improve their education.

By tackling these issues and providing a user-friendly solution, I believe this web app can make a positive impact on the lives of children and individuals with visual difficulties in Pakistan and all over the world.

Requirement Elicitation

To develop a web app that caters to the needs of individuals with visual difficulties, we conducted a thorough requirement elicitation process. The process involved identifying the essential features and functionalities required to address the challenges faced by people with visual difficulties. Also we searched for different blogs and articles on the internet written by professionals to get a thorough understanding of the things.


The first step in the requirement elicitation process was to identify the target audience and their specific needs. After conducting research and consulting with experts, I found that people with dyslexia, ADHD, and visual agnosia face significant challenges when it comes to learning and education. Therefore, the web app needed to be tailored to address these specific difficulties.

The next step was to identify the features and functionalities that would make the web app useful and effective.

The **primary features** that were identified during this process include:

1. **Diagnostic Tests:** The web app needed to include diagnostic tests to identify the specific difficulties that users face. These tests would help tailor the web app's features and functionalities to the individual's needs.
2. **Cognitive Enhancement Games:** The web app should include games that help improve cognitive abilities. These games would be tailored to the specific needs of individuals with visual difficulties.
3. **Personalized Learning Plans:** The web app should provide personalized learning plans that cater to the specific needs of each user.
4. **Progress Tracking:** The web app should include a scoring mechanism to track the progress of users. This feature would help users monitor their progress and motivate them to continue using the app.
5. **User Authentication:** The web app should include a login and sign-up page to authenticate users and store their data securely.
6. **Accessibility:** The web app should be designed to be accessible to individuals with visual difficulties. This includes features like text-to-speech and high contrast options.
7. **Easy Navigation:** The web app should have a user-friendly interface that is easy to navigate and use.

Once the features and functionalities were identified, the next step was to define the technical requirements of the web app. The web app would be developed using the MERN stack



(MongoDB, Express, React, and Node.js). It was chosen because it is the highly used stack in the industry right now . So to learn it we chose this stack

In conclusion, the requirement elicitation process played a crucial role in identifying the essential features and functionalities required to develop a web app that addresses the challenges faced by individuals with visual difficulties. The identified features, along with the technical requirements, formed the basis of the web app's design and development.



Implementation

Backend

For the backend implementation, the following steps were undertaken to ensure seamless integration and efficient handling of data:

Database Selection:

- MongoDB: Since there was no relational data involved, a NoSQL database was chosen. MongoDB was selected as the database management system.

Database Schema:

- The user data schema consists of the following fields: first name, last name, email, password, and booleans for difficulties such as ADHD, Dyslexia, and Visual Agnosia.
- An additional boolean field is present to indicate whether the user has taken the diagnosis test.
- For each difficulty, an array is used to store scores for three levels in the first three indices, with the fourth index representing the current level.

Database Connectivity:

- Mongoose: To establish a connection between the Express.js server and MongoDB, the Mongoose library was used. Mongoose simplifies data modeling and provides a straightforward interface for database operations.
- MongoDB Atlas: The web application utilized the MongoDB Atlas cloud service for hosting the database, ensuring reliable and scalable access to the data.

API Endpoints and Routes:

- Express.js: The backend framework chosen for this project was Express.js. It was used to create separate routes and define endpoints for various functionalities, such as registration, login, and dashboard.
- Each endpoint handles specific requests, allowing the frontend to communicate with the backend server effectively.



User Registration:

- During user registration, the form prompts the user to enter their first name, last name, email, and age.
- Data Validation: The entered data is validated using the Joi library, which ensures that the data matches the specified data types and meets the required constraints.
- Password Encryption: The bcrypt library is used to securely encrypt the user's password before storing it in the database.

User Authentication:

- JSON Web Tokens (JWT): Instead of traditional session management, JWT is employed for authentication purposes. When a user is successfully authenticated, a JWT is generated and signed with a private key.
- The signed JWT is then stored in the client's end, allowing for subsequent requests to be authenticated.

Dashboard:

- User Data Retrieval: On the dashboard, the user ID is extracted from the JWT using the `jwt_decode()` function.
- With the user ID, all relevant user data can be retrieved from the database and displayed on the dashboard, providing a personalized experience for each user.

List of Libraries Used:

- Mongoose (for MongoDB connectivity)
- Joi (for data validation)
- bcrypt (for password encryption)
- Express.js (for creating API endpoints and routes)
- JSON Web Tokens (JWT) (for user authentication)



Frontend

The frontend implementation of the web application focused on creating an intuitive and accessible user interface that caters to individuals with visual difficulties. The following steps were undertaken to ensure a smooth and engaging user experience:

React.js Framework:

React.js, a popular JavaScript library, was chosen as the frontend framework. It enables the creation of interactive and dynamic user interfaces, making it suitable for building responsive web applications.

Component-Based Structure:

The frontend was structured using a component-based approach. Each component represents a distinct part of the user interface and can be reused throughout the application. This modular structure allows for better code organization, maintenance, and reusability.

Landing Page:

The landing page serves as the entry point for users. It provides a visually appealing and informative introduction to the web application. The landing page highlights the key features, benefits, and instructions on how to get started.

Login and Signup Pages:

Separate login and signup pages were developed to facilitate user authentication and account creation. Users can enter their credentials or create a new account by providing necessary information such as name, email, and password.

Dashboard:

The dashboard serves as the central hub for users after logging in. It provides a personalized and informative overview of the user's progress, test results, and learning plan. The dashboard is designed to be easily navigable, allowing users to access different sections of the web application effortlessly.



Learning Plans Module:

The Learning Plans page is a dedicated section of the web application that allows users to create and manage personalized learning plans for addressing their specific difficulties. This feature empowers users to customize their learning experience and focus on areas they want to improve. The Learning Plans page offers the following functionality:

1. Creating Learning Plans:
 - Users have the ability to create new learning plans by providing a title and description for each plan. These plans can be tailored to target specific difficulties such as dyslexia, ADHD, or visual agnosia.
2. Managing Learning Plans:
 - The Learning Plans page provides an overview of all the existing learning plans associated with the user's account. Each plan is displayed with its title, description, and other relevant details. Users can easily manage their plans by viewing, editing, and deleting them as needed.
3. Adding Plans:
 - To add a new learning plan, users can click on the "Add Plan" button opening a form where we can use dynamically generated radio buttons to add plans
4. Removing Plans:
 - If a user decides they no longer need a particular learning plan, they can remove it from their list. This can be done by selecting the plan and clicking on a "Remove" or "Delete" button. A confirmation prompt may appear to ensure the user's intent before permanently deleting the plan.

The Learning Plans page provides users with the flexibility to create, customize, and manage their learning journey. By allowing users to add and remove plans, they have control over the content and focus of their learning experience. This feature empowers users to adapt their learning plans based on their progress, changing needs, and evolving goals.



Test and Diagnostic Module:

A dedicated module was created to administer diagnostic tests for identifying specific difficulties such as dyslexia, ADHD, and visual agnosia. Users can take these tests by filling in a multi-page form, and the results are stored in the backend for analysis and generating personalized learning plans.

Cognitive Enhancement Games:

The web application incorporates various cognitive enhancement games to improve cognitive abilities. These games are designed to be engaging, interactive, and specifically targeted towards addressing the difficulties faced by individuals with visual impairments.

Styling and User Interface Design:

Attention was given to the visual aesthetics and user interface design of the web application. A visually appealing and consistent design was maintained throughout the application, incorporating appropriate color schemes, typography, and layout to enhance the user experience.

By implementing these frontend components and considerations, the web application offers a user-friendly interface that caters to the specific needs of individuals with visual difficulties. The React.js framework, along with a component-based structure, facilitates efficient development, code reuse, and seamless navigation. With a focus on accessibility and a visually pleasing design, the frontend implementation complements the backend functionality, providing users with an engaging and inclusive experience.

List of Libraries used:

- Bootstrap (Styling)
- React (JSX Syntax)
- React-Router (used for Navigation)
- React-Router-DOM (used for State Variables)

Further Improvements

Given the time constraints of the project and the importance of helping individuals with visual difficulties, there are some areas where we can make additional improvements to provide a better experience and offer more activities to address their difficulties. Here are some suggestions:



Hosting:

We have planned to further improve this app and get it hosted on some domain preferably Eduaid.com

More Activities for Treatment:

We can add more activities that specifically target the difficulties individuals face. These activities could include exercises to improve memory, word recognition, and problem-solving skills. By including a wider variety of activities, users can have more options to work on their difficulties.

Audio-Based Learning:

For individuals with dyslexia or visual agnosia, we can introduce audio-based learning components. This means including audio versions of text-based content and adding features like text-to-speech. These additions can help improve reading comprehension and auditory processing skills.

Interactive Exercises for ADHD:

To address attention difficulties associated with ADHD, we can include interactive exercises. These exercises can focus on attention span, task switching, and impulse control. By making them interactive and engaging, users with ADHD can benefit from these activities.

Guided Meditation and Relaxation Techniques:

Including guided meditation and relaxation techniques can be helpful for individuals with visual difficulties who experience anxiety or stress. These techniques can promote relaxation, reduce anxiety levels, and contribute to overall well-being.

Peer Support and Community Features:

Adding features that encourage peer support and community engagement can be valuable. This could include discussion forums, chat functionalities, or online support groups within the web app. Such features allow users to connect with others facing similar challenges, share experiences, and provide support to one another.

Installation and Usage

Go to the GitHub repository at following link to find the complete source code
<https://github.com/uzair-7886/Edu-Aid>

Prerequisite:

You should have **Git** and **nodeJs** already installed on your computer

Steps:

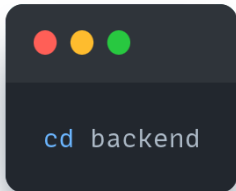
- Open the desired directory in terminal
- Clone the repo by typing the following command



```
git clone https://github.com/uzair-7886/Edu-Aid.git
```

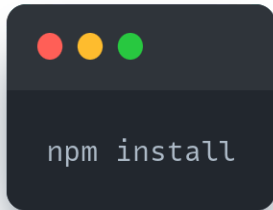
snappify.com

- Go to each separate directory and install the respective **node_modules** by typing the following commands as follows



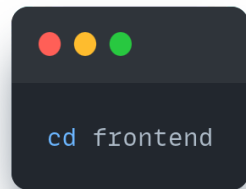
```
cd backend
```

snappify.com

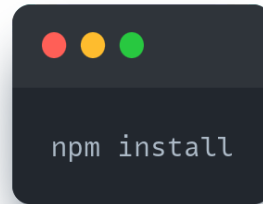


```
npm install
```

snappify.com

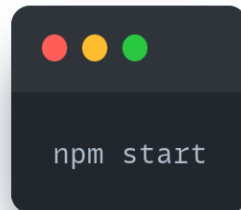


snapp*



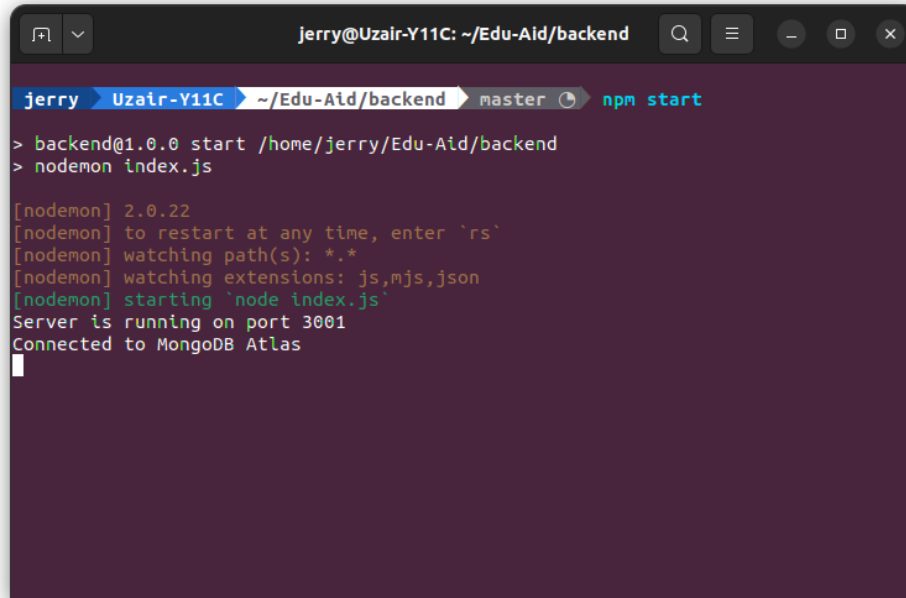
snapp*

- Now that relevant node_modules are installed you can start each server. Head back to the backend directory and type the following command in terminal to start the backend server at port **3001**



snapp*

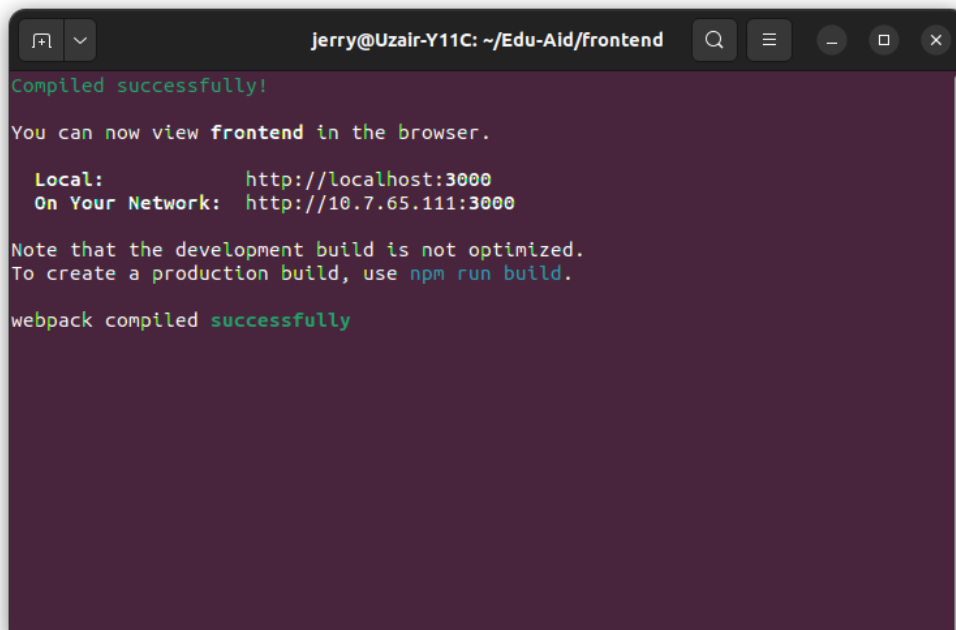
- You'll get the following message at your terminal screen



```
jerry@Uzair-Y11C: ~/Edu-Aid/backend master npm start
> backend@1.0.0 start /home/jerry/Edu-Aid/backend
> nodemon index.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server is running on port 3001
Connected to MongoDB Atlas
```

- Similarly do the same for frontend, write **npm start** and you'll get the frontend client starting at **port 3000** by default



```
jerry@Uzair-Y11C: ~/Edu-Aid/frontend
Compiled successfully!

You can now view frontend in the browser.

   Local:            http://localhost:3000
  On Your Network:  http://10.7.65.111:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```




Conclusion

In this project we learned a lot ,especially all the MERN stack we learnt by this project. We tried our level best to make it as better as possible but this is all that we could do.

Thanks for reading!