

A photograph of a person sitting at a desk, working on a laptop. They are holding a pen and writing in a notebook. On the desk are several books, papers, and a pair of glasses. The background is slightly blurred.

Final Year Project Presentation

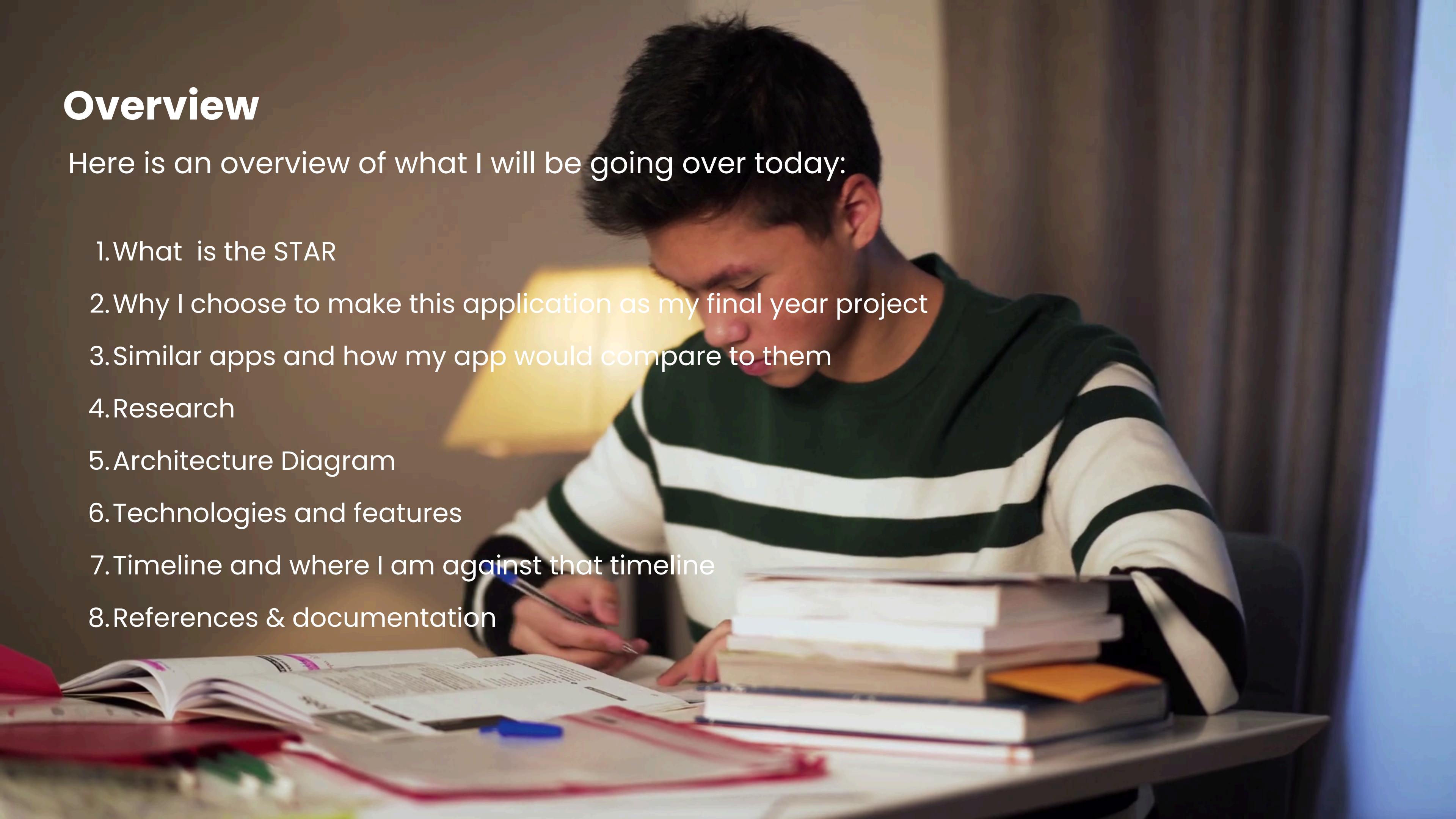
# **STAR: Student Task Assistant Resource**

**By Johnson Shogbaise**

# Overview

Here is an overview of what I will be going over today:

- 1.What is the STAR
- 2.Why I choose to make this application as my final year project
- 3.Similar apps and how my app would compare to them
- 4.Research
- 5.Architecture Diagram
- 6.Technologies and features
- 7.Timeline and where I am against that timeline
- 8.References & documentation



# What is the STAR

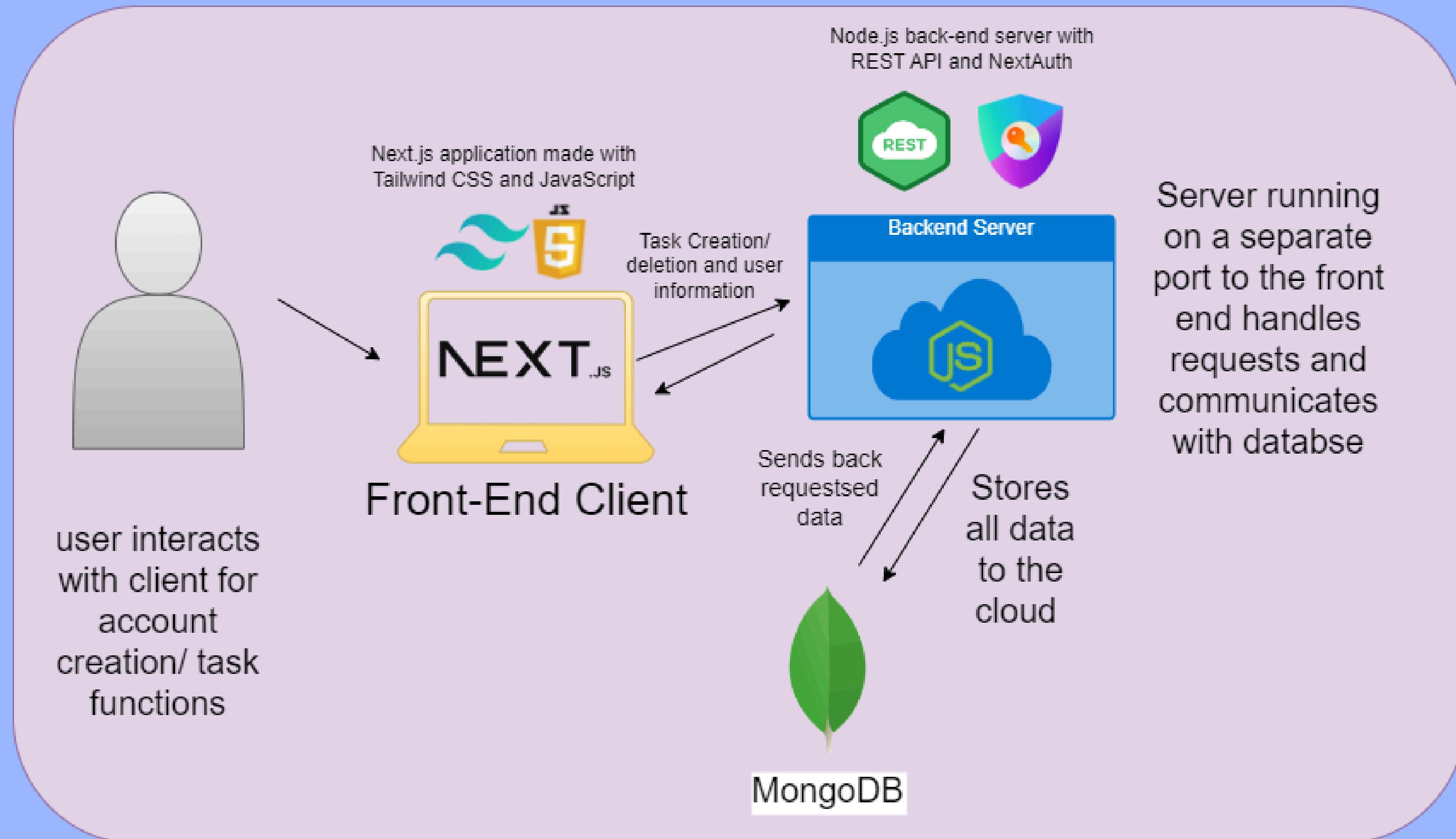
- The Student Task Assistant Resource or 'STAR' for short is essentially a todo list application aimed at making the life of college students easier, in regards to organisation.
- I came up with this idea after thinking of what I would want as a student that could help me achieve the most optimal work output and study I could achieve in school.



# Why did I choose this project

- There are already many todo list and task tracking applications out there on the market that could help improve my organisation skills.
- So why would I choose this project?
- Well I looked at the problems I face as a student:
  - Tracking submissions
  - Tracking changes in the school timetable
  - remembering what classes I have on a certain day and what time those classes start.
- Most todo list applications can track day to day task and even tasks that aren't due for a long time. But in my opinion, putting all the lecture times into a todo app calendar and have to go into that app everyday so I could see when my classes are is quite troublesome.
- So i wanted to make this app to make that process a little easier and more convenient

# Architecture Diagram



# Research

- With this project, i did of research on react technologies and looked at online resources such as documentation and youtube videos
- I have no prior experience with creating application front end and backend prior to this project, so most of the research i did was on the basics of this.
- I looked at the potential for using apis like the Google Calendar's API and OpenAI.
- All of this was tracked with monday project planner in Kanban format
- I researched helpful documentation on Next.js and Express and utilized these docs. throughout the creation of my project.



# Similar Applications



**Todoist**



**TickTick**



**Microsoft To Do**

# How these apps compare to the STAR

- All these applications have amazing features that make them different from each other and unique.
- But they are not aimed at students. They are more-so aimed at project groups
- With STAR, It would be made with features that would be more appealing to students and very easy to use



# Project Planning

For my project, i used Monday project planner. This played a huge roll in m project

The screenshot shows the Monday.com interface for the 'STAR' workspace. The main view is a table of tasks:

	Task	Status	Due date	Priority	Timeline
<input type="checkbox"/>	application design	Not Started			-
<input type="checkbox"/>	signup and login	Done	20-Mar		-
<input type="checkbox"/>	database creation	Done	10-Dec, 2023		-
<input type="checkbox"/>	page navigation 2	Done			-
<input type="checkbox"/>	Creating task editing and deletion fun...	Done	7-Dec, 2023		-
<input type="checkbox"/>	Logo Design	Done	22-Nov, 2023		-
<input type="checkbox"/>	UI Design	Done	23-Nov, 2023	Medium	w 18, '23 - Nov 21,
<input type="checkbox"/>	Creating the task addition function	Done	29-Nov, 2023		w 17, '23 - Nov 18,
<input type="checkbox"/>	Basic ToDo List Application w/ React	Done	16-Nov, 2023	Critical ⚡	w 13, '23 - Nov 14,
<input type="checkbox"/>	Project Proposal	Done	20-Oct, 2023		-
<a href="#">+ Add task</a>					

On the left sidebar, there are links for Home, My work, Main workspace, and a search bar. The top navigation bar includes 'See plans', 'Integrate', 'Automate', 'Invite / 1', and other account-related options.

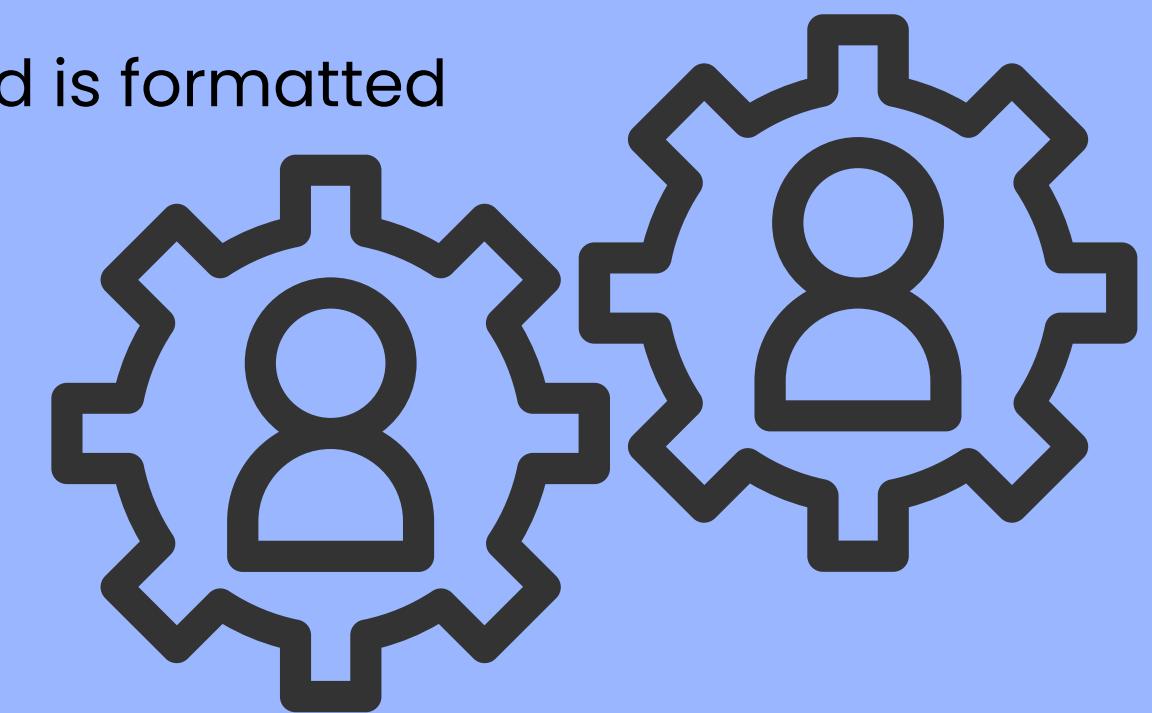
# Features and Technologies

- Features:
  - Adding, Editing, Deleting and Completing tasks.
  - Importing timetable, so the user can view their application on the app.
  - Task Prioritization & Status
  - Calendar
  - Categories.
  - User Creation and information protection
  - Session Creation
  - Database Transaction
- Technologies:
  - NextJS - TailwindCSS & JavaScript
  - Node.js - Express + REST API
  - Hooks - using useState to track state.
  - MongoDB Compass + Atlas



# Functionality

- A user enters a unique username, email and password to signup. if the user does not already exist, the user will not be created
- They are taken to the login page and must login. The user's information on signup is saved to the database and their password is protected
- The login and signup both call to different routes. Once the login is verified (i.e., "the user's email exists and the password is correct"), the user is pushed to the main page.
- A user can create a task, set a priority and put it under its appropriate category.
- The tasks are stored in the database and upon task update, the updates are saved to the database also.
- Tasks can also be deleted from the database when the trash icon beside each task is deleted.
- A timetable in .xlsx or .xls format can be inputted into the app and is formatted and displayed in the app



# Results

Login

Email

your-email

Password

password

NO SIGNUP

[Not a member? Sign up here!](#)

Signup

Username

your-username

Email

your-email

Password

password

NO SIGNUP

[Already signed up? Log in here](#)



test

25/04/2024, 00:42 p.m.

started

# Conclusion

- With all the planning, work and research done throughout the whole year, i was able to create a tool that students can utilize in their every day school live
- I learnt about th importance of planning, time managment and teamwork
- I wish to further develop this project in the future by opening avenues for a mobile phone application



# Thank you for listening



# Resources Used with this project

- Google Calendar API Overview (<https://developers.google.com/calendar/api/guides/overview>)
- Notifications API ([https://developer.mozilla.org/en-US/docs/Web/API/Notifications\\_API/Using\\_the\\_Notifications\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API/Using_the_Notifications_API))
- Next.js docs, <https://nextjs.org/>
- Express.js documentation, [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs)
- Open AI Documentation (<https://platform.openai.com/docs/overview>)