END-TO-END PROJECT ON:

TO-DO LIST APP

A PROJECT REPORT BY:

ILYAZ AHMED

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ABSTRACT

A to-do list is a list of tasks that need to be completed, typically organized in order of priority. It is one of the simplest solutions for task management and provides a minimal and elegant way for managing tasks a person wishes to accomplish.

Our aim is to design a simple and elegant website for people to keep a track of the status of their tasks. Making a to-do list is an easy and important task that everyone should do. The immense satisfaction that one gets when completing the task and marking it on the list are incomparable. Moreover, creating a list of tasks ensure you don't miss out on anything. It's a scientific fact that when you write the tasks that you need to complete, you are even more motivated to complete it. With this in mind, come to build a platform which will help people create their own task list. With the help of modern tools and technologies, it strives to build a minimal and efficient to-do list which minimizes distractions and helps people achieve task management with ease and without hassle.

GRAPHICAL ABSTRACT

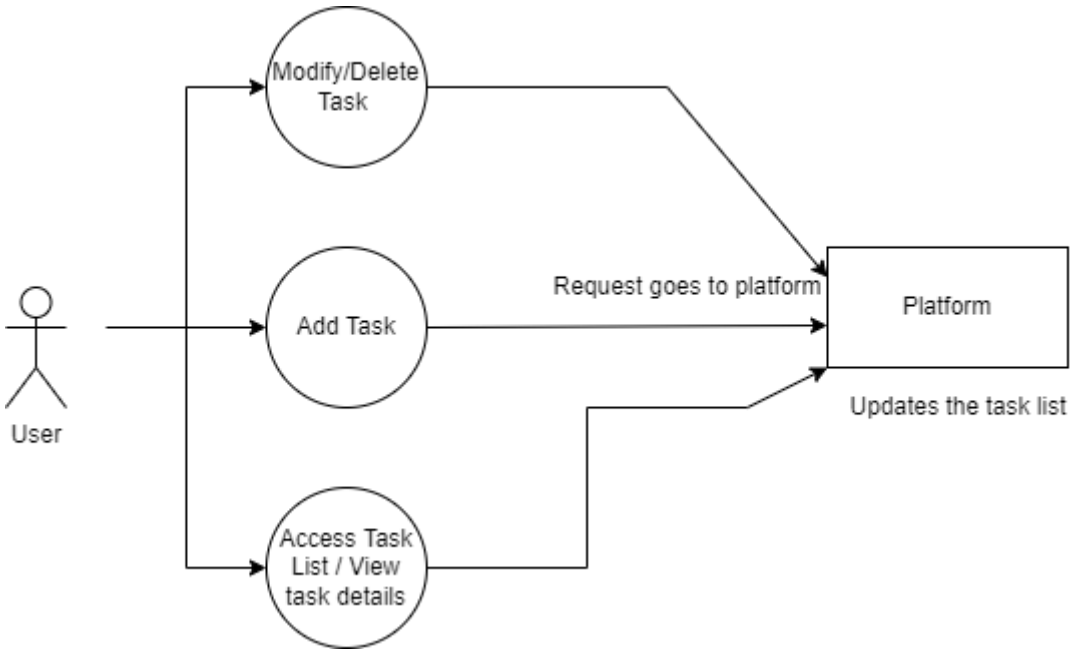

Fig. Graphical Abstract 1.1

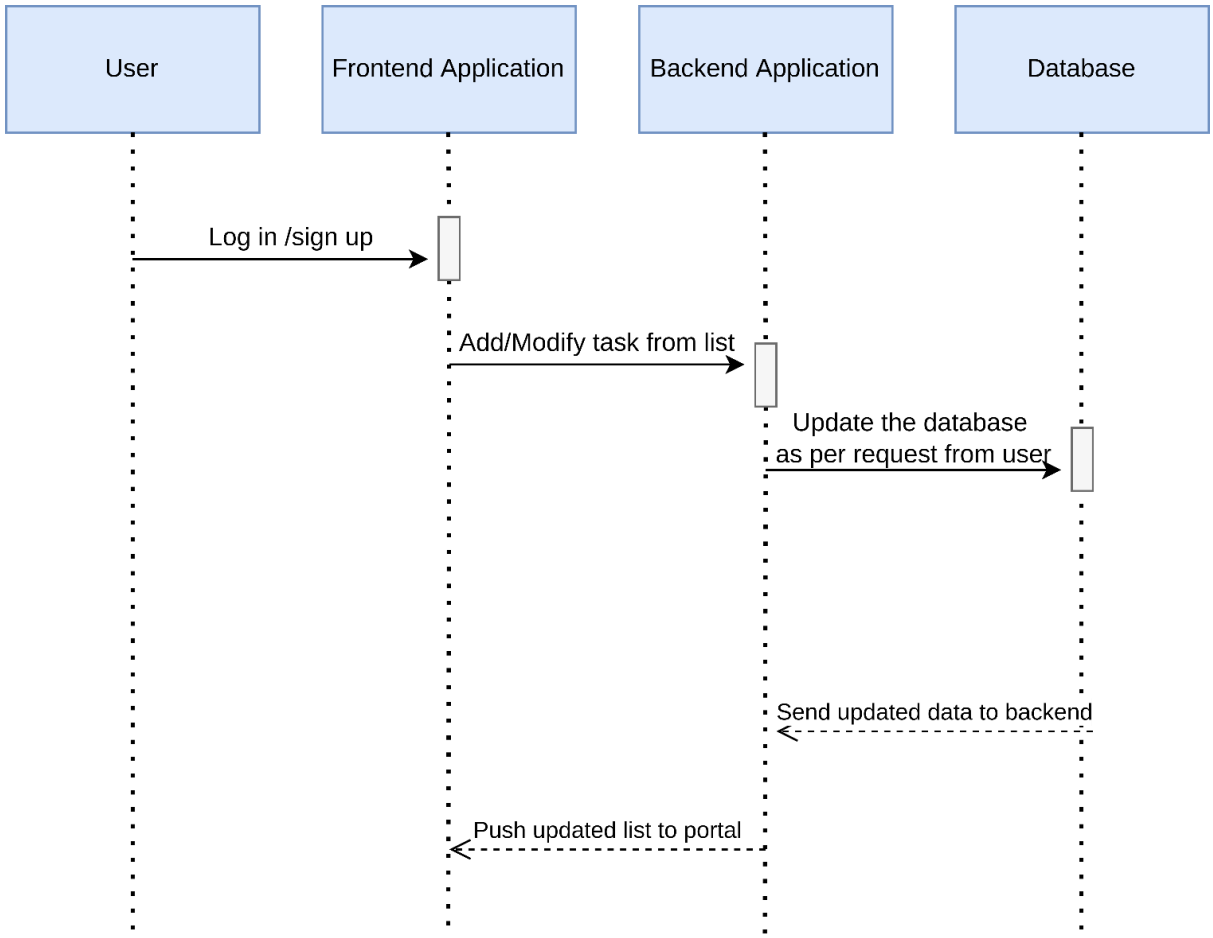

Fig. Graphical Abstract 1.2

ABBREVIATIONS

Abbreviations	*Definitions*
1. 3D	Three Dimensional
2. API	Application Programmable Interface
3. B.E.	Bachelor of Engineering
4. CSS	Cascading Style Sheets
5. DFM	Design for Manufacturability
6. DOM	Document Object Model
7. Dr.	Doctor
8. ES6	ECMAScript6
9. ESA	Entertainment Software Association
10. Etc.	Etcetera
11. Fig.	Figure
12. FoS	Factor of Safety
13. JS	JavaScript
14. JSX	JavaScript XML
15. NAICS	North American Industry Classification System
16. PC	Personal Computer
17. PHP	Hypertext Pre-processor
18. SASS	Syntactically Awesome Style Sheets
19. SEMVER	Semantic Versioning
20. SDG	Sustainable Developments Goals
21. SF	Safety Factor
22. SVS	Scalable Vector Graphics
23. UI	User Interface
24. U.S.	United States
25. W3C	World Wide Web Consortium

SYMBOLS

Symbols	*Definitions*
1. %	Percent
2. *	Asterisk
3. →	Right Arrow
4. ×	Multiplication Symbol

CHAPTER 1: INTRODUCTION

A to-do list is a simple prioritized list of the tasks a person must complete. People make a list of everything they need to do, ranked according to priority from the mostcritical task at the top to the least critical task at the bottom

A few of the features of a good to-do list application include:

- Plan and execute simple actions.
- Prioritize, manage, and reason about tasks.
- Record notes, action items and ideas.

To-dos are the tasks or the atomic entities that make up a to-do list. To-dos are made quickly, the bulk of them do not specify the work; instead, they are typically just comprehensive enough to serve as a valuable indicator. To be sure, to-do terminologylike "Groceries" or "Car Wash" is frequently grammatically correct. Because the signal is so quick, it is only useful for a short period of time while the task is remembered. In certain cases, a simple item like a stack is enough to recall the job without the need for a note.

There are clear immediate implications to adding a to-do list to a person's productivity system. The functionalities provided by a good to-do list application/system help declutter the user's mind as their pending tasks are recordedsafely and they won't be forgotten.

The To-do list project is a user-friendly website which helps them to keep a track of their tasks. It is a simple site which requires no sign-in/log-in or any personal details but still records your task, mark the completed tasks, and stores them even if you visitthe site after a few days.

Traditionally, they're written on a piece of paper or post it notes and acts as a memoryaid. As technology has evolved, we have been able to create a to-do lists with excel spreadsheets, word documents, to-do list apps, Microsoft to do and google to do list to name a few. You can use to do list in your home and personal life, or in the workplace.

The Benefits of Using a To Do List

1. **Improves your memory**: A to do list acts as an external memory aid. It's only possible to hold a few pieces of information at one time. Keep a to do list and you'll be able to keep track of everything, rather than just a few of the tasksyou need to do. Your to do list will also reinforce the information, which makesit less likely you're going to forget something.
2. **Increases productivity**: A to do list allows you to prioritize the tasks that are more important. This means you don't waste time on tasks that don't require your immediate attention. Your list will help you stay focused on the tasks that are the most important.
3. **Helps with motivation**: To do lists are a great motivational tool because you can use them to clarify your goals. You can divide your long-term goal into smaller, more achievable short-term goals and as you tick each one off your list, your confidence will increase.

1.1 TASK MANAGEMENT

From an articulation standpoint, task management entails more than just organizing virtual and physical collections and scheduling activities.

Recent research has begun to address the problem of generic task management in the context of email. This development is hardly surprising, given that many digital device users are overloaded by the number of chores done through email. According to this research, any successful productivity tool must be tightly connected with emailfunctionalities. Recent researches looked at task management strategies more generally because email and related technologies are unlikely to be the whole picture.

1.2 To-do List as a solution to task management

The concept of to-do list has existed for a very long time and it is one of the primary methods for management of tasks, use of a to-dos as a reminder system, to-dos as a system for note management, etc. In the simplest and most primitive form, a to-do listcan be implemented on a pen and paper as a checklist of items which can be crossed of or ticked against when completed.

This can be further extended to calendars, by writing tasks against dates where the dates can also act as deadlines for particular tasks. Other possible extensions of to-dolists can be on whiteboards, journals, text editors.

The functionalities of to-do list naturally evolve to perfectly fit web applications andapplications on digital devices. Equipped with modern tools and technologies, engineers can build an application to create a minimal and powerful application that can help boost productivity without loss of focus and attention.

With the computing power and persistence of modern devices and databases, losingtrack of tasks will not be a problem people will have to face anymore and they can rest assured, only focused on the tasks they will have to accomplish as with moderntechnology and the power of digital devices, integration will be seamless and tasks can be synced across multiple devices all at once, without any hassle.

Essential Functionalities:

- A variety of methods for examining and managing to-dos that go beyond listsand mirror the advantages of current resources.
- The inconvenient property, such as when it becomes the default spot foreveryday tasks where reminders can be satisfied.
- Immediately turns on, allowing for quick input and clear view. Conventionalto-do lists are commonly abandoned due to slow, time-consuming input and weak output.
- No formal job description, classification, or decomposition is required from users, and any degree of abstraction for atomic task entries must be permitted.
- A mechanism for dealing with stale, low-priority to-dos that are becoming lesslikely to be performed but have not been explicitly deleted.

1.3 GANTT CHART

Name of the task	Start date	End date	Duration
Project kick off	2/14/2022	2/15/2022	1
initial design	2/16/2022	2/20/2022	4
making of prototype	2/21/2022	2/28/2022	7
Research	03/01/2022	03/07/2022	6
Adding functionality to prototype	03/08/2022	03/14/2022	6
Initiating development	03/15/2022	03/20/2022	5
Finalizing design	03/21/2022	03/24/2022	3
Developing prototype	03/25/2022	03/28/2022	3
Testing functionality	03/29/2022	03/31/2022	2
Implementing the solution	04/01/2022	04/06/2022	4
Initiating integration of modules	04/07/2022	04/12/2022	5
Finalizing essential functionalities	04/13/2022	04/20/2022	7
Planning addon functionalities	04/21/2022	04/26/2022	5
Implementing addon functionalities	04/27/2022	04/30/2022	3
testing of the project	05/01/2022	05/06/2022	5
implement improvements and debugging	05/07/2022	05/10/2022	3

Table 1.1

1.4 START DATE AND DURATION

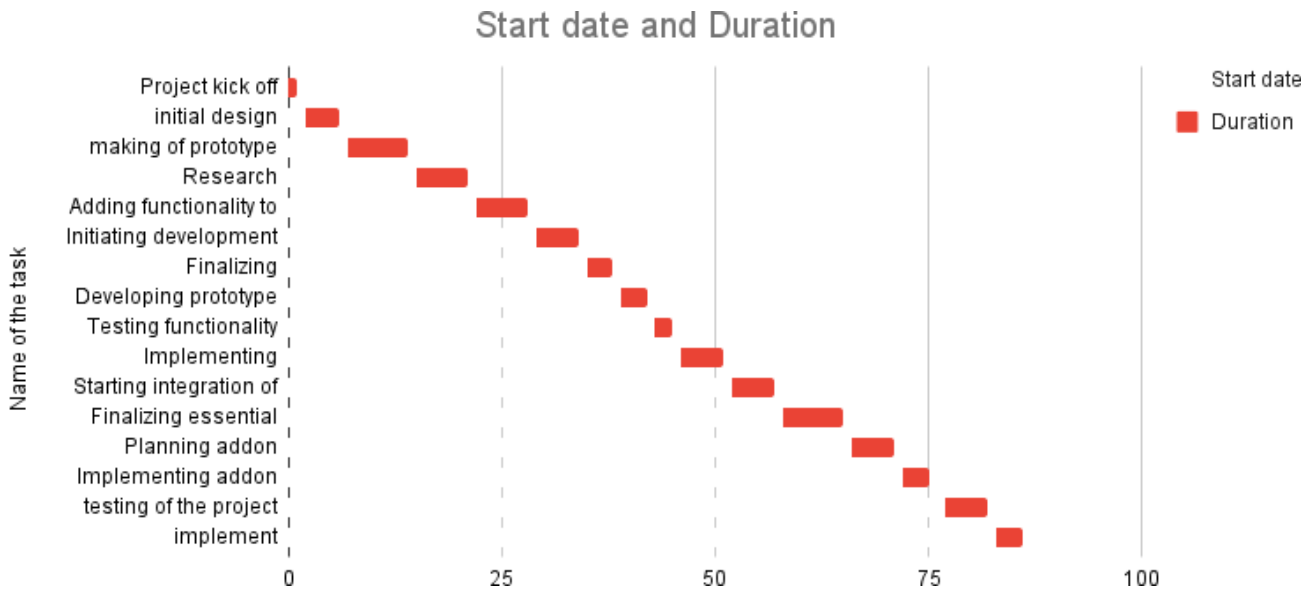

Fig. 1.3

CHAPTER 2: LITERATURE SURVEY

As discussed earlier, productivity and task management entails more than justorganizing virtual and physical collections and scheduling activities.

Recent research has begun to address the problem of generic task management in the context of email. This development is hardly surprising, given that many digital device users are overloaded by the number of chores done through email. According to this research, any successful productivity tool must be tightly connected with emailfunctionalities. Recent researches looked at task management strategies more generally because email and related technologies are unlikely to be the whole picture.

2.1.1 Evolution of to-do list

The idea of plan for the day has existed for quite a while and it is one of the essentialtechniques for the board of assignments, utilization of a tasks as an update framework, tasks as a framework for note the executives, and so on. In the least difficult and most crude structure, a plan for the day can be executed on a pen and paper as an agenda of things which can be crossed of or ticked against when finished.

This can be additionally reached out to schedules, by composing undertakings againstdates where the dates can likewise go about as cutoff times for specific assignments. Other potential augmentations of plans for the day can be on whiteboards, diaries, text editors.

The functionalities of plans for the day normally develop to fit web applications and applications on advanced gadgets flawlessly. Furnished with current apparatuses and advancements, specialists can fabricate an application to make a negligible and strongapplication that can assist with supporting efficiency without loss of concentration and consideration.

With the processing power and steadiness of current gadgets and data sets, forgetting about assignments won't be an issue individuals should confront any longer and theycan have confidence, just centered around the errands they should achieve similarly as with present day innovation and the force of advanced gadgets, combination willbe consistent and undertakings can be adjusted across different gadgets at the sametime, with next to no problem.

2.1.2 Essential Functionalities

1. An assortment of strategies for analysing and figuring out how tasks that gopast records and mirror the upsides of current assets.
2. The awkward property, for example, when it turns into the default spot forordinary errands where updates can be fulfilled.
3. Promptly turns on, taking into consideration fast information and clear view.Regular plans for the day are ordinarily deserted because of slow, tedious information and frail result.
4. No conventional set of working responsibilities, order, or deterioration is expected from clients, and any level of reflection for nuclear errand sectionsshould be allowed.
5. An instrument for managing old, low-need tasks that are turning out to be lessinclined to be performed yet have not been expressly erased.

2.1.3 Integration with other technologies

A To-do list is a simple and somewhat independent entity, which makes it easy tointegrate with related technologies to help create a better system of productivity overall.

Various studies have pointed towards and shed light on how well e-mail as a technology can be seamlessly integrated with to-do list technologies and the synergyof these two technologies help boost productivity greatly as e-mail and social media have now become an essential technology that people need to have access to in orderto be a productive member of any organization or institution as these have now become an essential part of everyday life.

Collaborative task management is also one of the problems that can be solved througha well designed and engineered to-do list application such as Trello, a web based, Kanban-style list application which helps teams organize their tasks and stay in sync when working collaboratively on a project.

Another example of a well-integrated to-do list application is Google Tasks, a to-do list application which is completely synchronized with other Google and third-party services such as Google Calendar, Mail, Clock, etc. Such well-integrated applicationshelp build a highly effective productivity system for a user.

2.1.4 Gamification

Gamification is the process of adding game elements into non-game environmentssuch as a website, productivity systems, or corporate intranet. Gamification encourages consumers, employees, and partners to collaborate, contribute, and interact by including them in games.

As shown by studies in Kappen, adding an element of gratification through gamification boosts engagement of users to platforms, even when it comes toproductivity.

2.2 PROBLEM DEFINITION

2.2.1 INTRODUCTION

The current state of task management through currently available to-do listapplications is a hotch-potch, to say the least. The highly available and most used to-do list applications are heavily bloated and provide unnecessary levels of integrations which are usually not required and clutter a user's productivity system.

A lot of good to-do list applications are not free to use and usually run ads to generate revenue, which is a huge negative point when it comes to productivityapps. Ones which are run by large companies are usually trying to pull users towards their own app ecosystem through non-sensical integrations and bloatware.

Some of the most glaring issues of to-do list productivity applications are discussed in this chapter as we try to identify the problems and shortcomings ofcurrently available solutions and build our project to overcome those shortcomings.

2.2.1 UNNECESSARY INTEGRATIONS

For most of the to-do list applications available to use, they are not independententities or applications. Most of these applications are usually meant to be usedwith other applications to form a "productivity system" where a user's calendar, clock, mail, notifications, etc. are all linked to their to-do list. While this may be desirable to most people and may work to improve productivity forsome, it is very often more distracting than useful and over-integration of applications often leads to sensory overload and overwhelming frustration.

Integrations are very often counter-productive and there should exist solutions that are absolutely independent entities, free of all clutter and meant to fulfill a simple and minimal purpose. Unfortunately, not many solutions like this exist currently.

2.2.3 PRODUCTIVITY APPS AS PAIDSERVICES

Ironically, most good to-do list applications are not free-to-use and almost always have some in-app purchase or subscription model to generate revenue.

While it is understandable that capital is required to maintain and provide someservices to users, a lot of those services are not necessary to most users, whoare generally students, and yet they have to pay for it.

This is one of the prominent problems with to-do list applications as very often all people need is a simple app to take notes and record tasks without payingfor it and that is all a good to-do list application needs to offer, a simple and minimal application to serve the mentioned functionalities to the user.

Most of any productivity application's user base is composed of students and they are usually the type of users with the least funds to expend, so having a minimal to-do list application can help the student demographic greatly.

2.2.4 RUNNING ADVERTISEMENTS TO GENERATE REVENUE

Although less prominent, it is not rare to see productivity applications running advertisements to generate revenue. Ads are the biggest distractions and a bane to productivity so although the developers can earn through ads, it is a bane to the users and very often they don't have a choice, but to keep using the same application with ads, which leads to a dip in productivity and focus.

2.3 OBJECTIVE (OF PROJECT)

To-do lists offer a way to increase productivity, stopping you from forgettingthings, helps prioritize tasks, manage tasks effectively, use time wisely and improve time management as well as workflow.

Making a to-do list is an easy and important task that everyone should do. Theimmense satisfaction that one gets when completing the task and marking it onthe list is incomparable. Moreover, creating a list of tasks ensures you don't miss out on anything. It's a scientific fact that when you write the tasks that you need to complete, you are even more motivated to complete it.

With this in mind, we come to build a platform which will help people createtheir own task list. The main goals we hope to accomplish with this project include:

2.3.1 MINIMAL INTERFACE

The primary focus of our application on the user interface aspect is minimalism. It should only provide the basic functionalities of a to-do list andnot anything unnecessary as that is a hinderance to focus and productivity of auser.

2.3.2FREE TO USE

The to-do list application build for this project is completely free to use to anyand everyone. The only requirement of this application is to have a device capable of running a web browser.

2.3.3 OPEN SOURCE

To-do list is open source and its code repository is publicly available on GitHub. It is open to contributions and improvements by anyone who wishes tocontribute to it. It is also free for modification and personal use.

A brief overview of open-source projects:

Open-source software is software with source code that anyone can inspect, modify, and enhance. "Source code" is the part of software that most computerusers don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works.

Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't alwayswork correctly.

2.3.4 AD-Free

There is no incentive for generation of revenue through this project so it willforever remain ad-free and clutter free.

2.3.5 Efficient

Very few resources are required to store and run the application.

2.3.6 CONCLUSION

The to-do list developed for this project is meant to be a minimal, free andopen-source application that can help improve the productivity of a user without taking anything. It aims to solve at least some of the problems discussed earlier in this report.

CHAPTER 3: DESIGN FLOW/PROCESS

Concept Generation, Evaluation & Selection of Specifications/Features, Design Constraints– Regulations, Economic, Environmental, Health, manufacturability, Safety, Professional, Ethical, Social & Political Issues considered in design, Analysis and Feature finalization subject to constraints, Design Flow (at least 2 alternative designs to make the project), Best Design selection (supported with comparison and reason) and Implementation plan ((Flowchart /algorithm/ detailed block diagram)).

3.1 CONCEPT GENERATION

In order to do manage tasks, the concept of to-do lists was formulated to provide a simple, hassle-free solution to the problem of taking short notes and recording the daily tasks that a person needs to complete. We extend this concept and implement it using modern tools and technologies.

3.2 EVALUATION & SELECTION OF SPECIFICATIONS/FEATURES

This project is divided into three main components: the frontend, backend and database. The frontend is the medium through which the user can interact with the application and utilize the necessary functionalities of the application. The backend isa two-way bridge between the database and the frontend. It ensures security, transfer and integrity of data that flows from the database to the frontend of the application.

The database is basically a store of all the information a user wishes to persistently store. It is responsible for storage, persistence, integrity and retrieval of data. The technologies used for the implementation of this project are: ReactJS for frontend, Node.js for backend and MongoDB for database as the primary focus of this projectis simplicity.

3.3 DESIGN CONSTRAINTS

3.3.1 ECONOMIC

To-do list is free to use. However, damage to a device's software and/or hardware due to improper modification of this application is always a possibility.

3.3.2 ENVIRONMENTAL

The use of To-do list does not generate any environmental concerns. The possible causes ofany problems may be from the hardware not being disposed properly which would lead to e-waste generation.

3.3.3 HEALTH

There are no health constraints. The only issues and problems that may arise might be due to the devices and hardware that the user uses. In case of any such situation or in the event of any problem or emergency in which the user feels biological discomfort, it is suggested toimmediately seek professional medical assistance.

3.3.4 MANUFACTURABILITY

As To-do list is digital software, there seem to be less issues with manufacturability.

3.3.5 SAFETY

To-do list it does not guarantee the physical safety of users. In case of intimidation orunderhanded malicious extortions, it is advised to seek police aid.

3.3.6 PROFESSIONAL

With the constant development in new technologies revolving around to-do list applications, new professional services are required to be developed as well. However, these will heavily depend on the regulations from the government.

3.4 PROFESSIONAL, ETHICAL, SOCIAL & POLITICAL ISSUES CONSIDERED IN DESIGN

There is no professional, ethical, social or political issues revolving aroundthe design of our to-do list application.

3.5 ANALYSIS AND FEATURE FINALIZATION SUBJECT TO CONSTRAINTS

The features of the to-do list applications include:

1. Viewing all the tasks in a user's to-do list
2. Marking a task as completed, changing its priority to 0
3. Tasks are ordered by priority of importance
4. Deletion of tasks
5. Attributing due dates to tasks
6. Tasks are highlighted when they are close to due or overdue.
7. Adding tasks to a user's to-do list

3.6 DESIGN FLOW

3.6.1 Final Design:

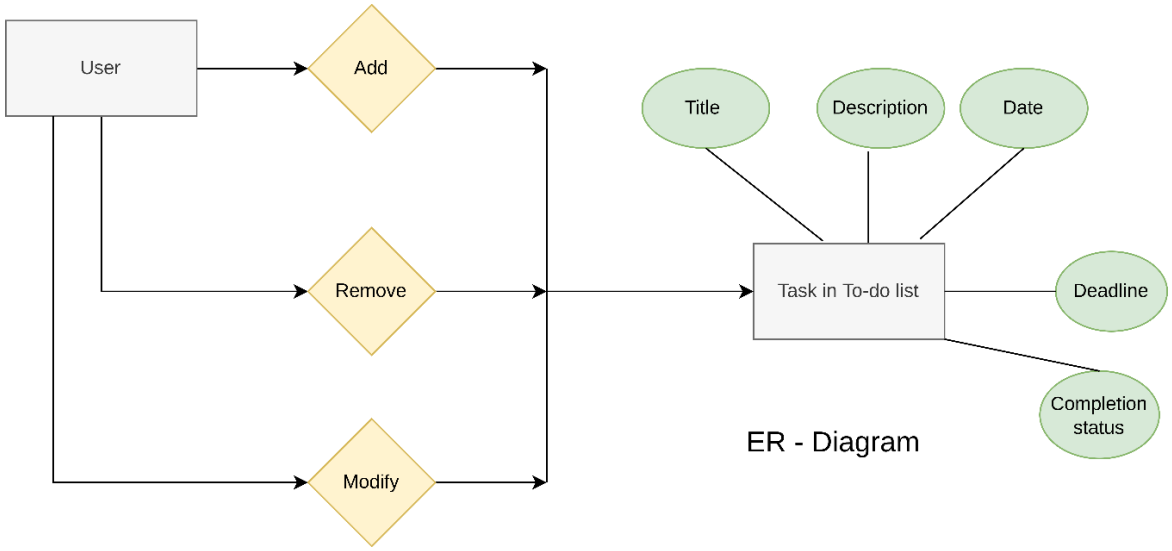

Fig: Entity-Relationship Diagram

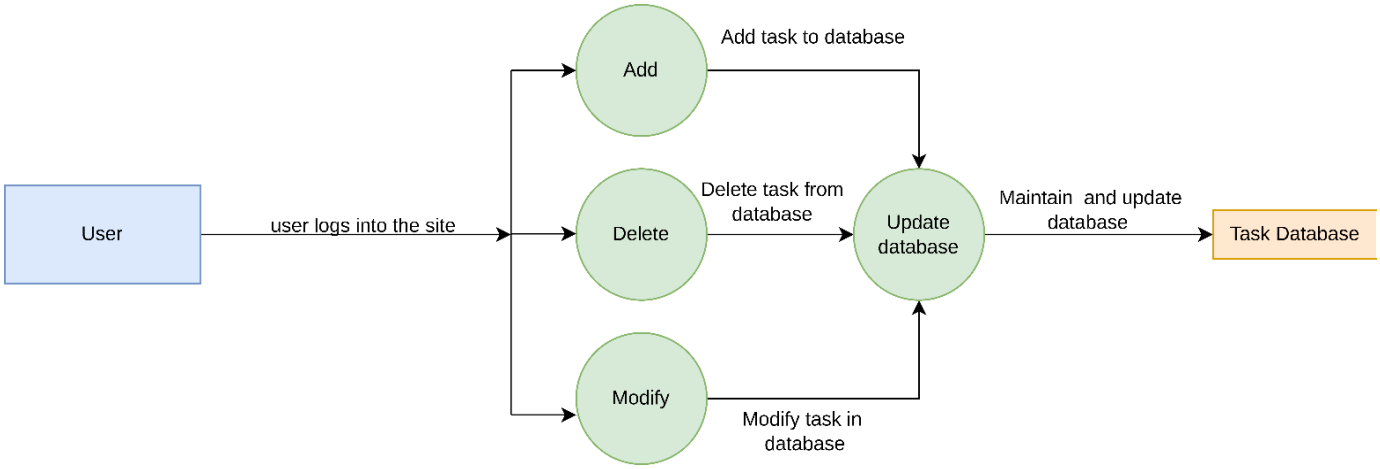

Fig 3.2: Data flow diagram

3.6.2 Alternative designs:

- One possible alteration to the design of the project is to add location-based tasktracking to remind a user to complete a task triggered by the location of the

  user. This design was not chosen for the project because location services consume significantly more resources and thus it would render the application inefficient for mobile devices.

- E-mail based reminders can also be seen as alternative design for the project but were not implemented because they require access to a user's email and details, which could possibly risk user anonymity and be a potential securityrisk without adding proper security middleware to the project.

3.7 BEST DESIGN SELECTION

The final design chosen for the implementation of this project was to divide it into three main components: the frontend, backend and database. The frontend is the medium through which the user can interact with the application and utilize the necessary functionalities of the application. The backend is a two-way bridge between the database and the frontend. It ensures security, transfer and integrity of data that flows from the database to the frontend of the application. The database is basically a store of all the information a user wishes to persistently store. It is responsible for storage, persistence, integrity and retrieval of data. The technologies used for the implementation of this project are: ReactJS for frontend, Node.js for backend and MongoDB for database as the primary focus of this project is simplicity.

The main functionalities that were added to this project are:

1. Viewing all the tasks in a user's to-do list
2. Marking a task as completed, changing its priority to 0
3. Tasks are ordered by priority of importance
4. Deletion of tasks
5. Attributing due dates to tasks
6. Tasks are highlighted when they are close to due or overdue
7. Adding tasks to a user's to-do list

3.8 IMPLEMENTATION PLAN

The implementation plan and timeline shown through a Gantt chart:

Name of the task	**Start date**	**End date**	**Duration**
Project kick off	2/14/2022	2/15/2022	1
initial design	2/16/2022	2/20/2022	4
making of prototype	2/21/2022	2/28/2022	7
Research	03/01/2022	03/07/2022	6
Adding functionality to prototype	03/08/2022	03/14/2022	6
Initiating development	03/15/2022	03/20/2022	5
Finalizing design	03/21/2022	03/24/2022	3
Developing prototype	03/25/2022	03/28/2022	3
Testing functionality	03/29/2022	03/31/2022	2
Implementing the solution	04/01/2022	04/06/2022	4
Initiating integration of modules	04/07/2022	04/12/2022	5
Finalizing essential functionalities	04/13/2022	04/20/2022	7
Planning addon functionalities	04/21/2022	04/26/2022	5
Implementing addon functionalities	04/27/2022	04/30/2022	3
testing of the project	05/01/2022	05/06/2022	5
implement improvements and debugging	05/07/2022	05/10/2022	3

Table 3.1

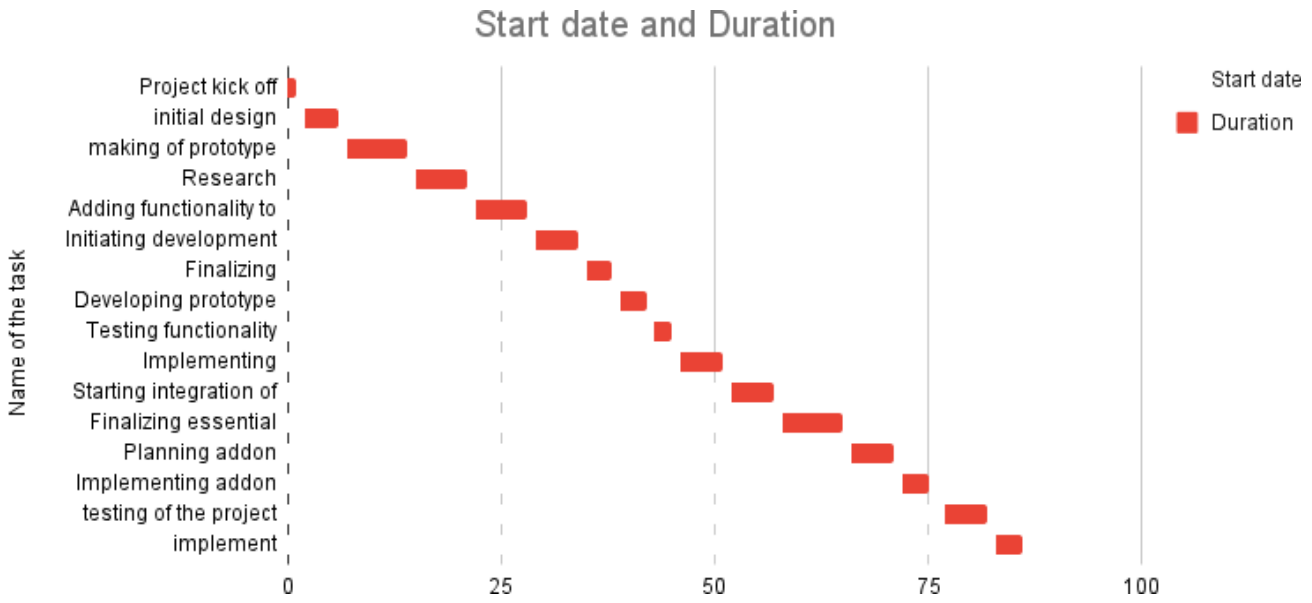

Fig 3.3 Gantt Chart

CHAPTER 4: RESULT ANALYSIS AND VALIDATION

4.1 The technologies that are used to implement this project are:

4.1.1 ReactJS:

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based onUI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page, mobile, or server-renderedapplications with frameworks like Next.js. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

We chose React as the framework for the frontend of the application because it is simple to implement and work with and the to-do list app can be elegantly designed as a single page application (SPA).

4.1.2 Node.js:

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

4.1.3 MongoDB:

MongoDB a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License (SSPL).

MongoDB has great compatibility with Node.js and the Mongoose library provides great integration between the server and database in Node.js which makes it very convenient to access and modify the database, which is a great advantage especially for small scale projects.

4.1.4 Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source- code editor made by Microsoft for Windows, Linux and macOS. Features include supportfor debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users canchange the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

4.1.5 Git version control system

Git is a software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals includespeed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

4.2 Use-case diagram

.

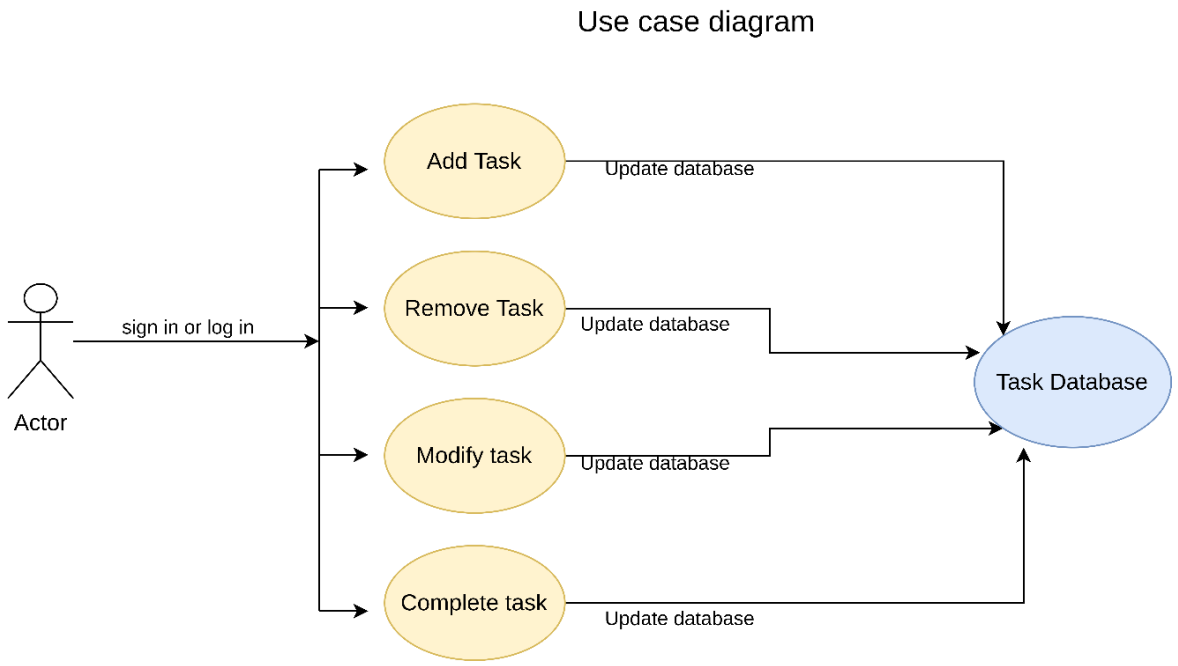

Fig 4.1

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 DEVIATIONS FROM EXPECTED RESULTS AND WAYAHEAD

Some unexpected deviations and resulting design changes during the development of this project include:

- The initial plan was to use an SQL database instead of MongoDB but the Object Relational Mapping (ORM) library we planned to use (Sequelize) with Node.js had issues performing queries on local PostgreSQL database so we tried implementing the project with MongoDB instead and decided to go on with it.
- The feature to rank and list tasks by their priority was implemented towards the end of the project as we had initially planned to classifytasks into two lists only based on whether the tasks were completedor pending.
- Several implementation problems did not have any solutions that could be solved by directly using third-party libraries so they had tobe solved by manually implementing the functionalities.
- Improvement in security and integrity
- Integration with other applications such as calendar, mail, etc.
- Implementation of authentication
- Hosting the web application online to make it accessible to moreusers

5.2 REFERENCES

1. Staff, C. A. C. M. (2016). React: Facebook's functional turn on writing JavaScript. *Communications ofthe ACM*, *59*(12), 56-62.
2. Kvalheim, C. (2015). The Little Mongo DB Schema Design Book. *The Blue Print Series*.
3. Khedkar, S., Thube, S., Estate, W. I., & Naka, C. (2017). Real time databases for applications. *International Research Journal of Engineering and Technology (IRJET)*, *4*(06), 2078-2082.
4. Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance networkprograms. *IEEE Internet Computing*, *14*(6), 80-83.
5. Syed, B. A., & Bean, M. (2014). *Beginning Node. js* (pp. 181-182). New York City: Apress.
6. Morales-Morell, A. (2001). *Usability aspects of a location-aware TODO list application*. University ofPuerto Rico, Mayaguez (Puerto Rico).
7. Gaspar, F., Taniça, L., Tomás, P., Ilic, A., & Sousa, L. (2015). A framework for application-guided taskmanagement on heterogeneous embedded systems. *ACM Transactions on Architecture and Code Optimization (TACO)*, *12*(4), 1-25.
8. Lin, C. Y., Hung, M. T., & Huang, W. H. (2012, September). A location-based personal taskmanagement application for indoor and outdoor environments. In *2012 15th International Conference on Network-Based Information Systems* (pp. 582-587). IEEE.
9. Bellotti, V., Ducheneaut, N., Howard, M., & Smith, I. (2002). Taskmaster: recasting email as taskmanagement. *PARC, CSCW*, *2*.
10. Unger, T., & Bauer, T. (2008). Towards a Standardized Task Management. In *MultikonferenzWirtschaftsinformatik* (pp. 443-444).