

**Khulna University of Engineering & Technology, Khulna.**

**Department of Computer Science and Engineering**

Project title: **Task Management**

Prepared by:

| Saurav Dev | Simon Walter Tudu |
| --- | --- |
| 1807027 | 1807121 |
| Department of Computer Science and Engineering, KUET | Department of Computer Science and Engineering, KUET |

**Supervisor:**

**Prottoy Saha**

**Assistant Professor**

Department of Computer Science and Engineering Khulna University of Engineering & Technology Khulna -9203, Bangladesh.

Signature



**Acknowledgement**

With heartfelt gratitude, we would like to express our deepest appreciation to all those who provided us the possibility to complete this project on “Task Management” as a part of CSE 3200: System Development Project. We are thankful to our supervisor **Prottoy Saha** sir who helped and guided us to complete this project successfully. His contribution was remarkable.

# Table of Contents



## Chapter Title Page No.

**Chapter 1 Introduction**

|  | 1. Background |  |  | 1 |
| --- | --- | --- | --- | --- |
| 1. Objectives |  |  | 2 |
|  | 1. Platform |  |  | 2 |

**Chapter 2 The Interface**

* + - 1. Login 4
      2. Register 5
      3. Task 6
      4. Dashboard 7
      5. Completed Task 7
      6. Progress Task 8
      7. Canceled Task 8
      8. Profile 9
      9. Mobile Views 9

**Chapter 3 Methodology**

|  | 3.1 | Data Collection and Description | 11 |
| --- | --- | --- | --- |
| 3.2 | MongoDB Database | 12 |
| ` | 3.3 | Data manipulation | 12 |
|  | 3.4 | Data in MongoDB Views | 13 |

**Chapter 4 Flow Chart**

* 1. [Flow Char 14](#_2xcytpi)

**Chapter 5 Limitations, Future Work and Contributions**

* 1. [Limitations 15](#_1ci93xb)
  2. [Future Work 15](#_3whwml4)

5.3 Conclusion 15

[**References**………………………………………………………………………………….16](#_2bn6wsx)

# List of Figures



## Figure Page

Fig 2.1: Login page 4

Fig 2.2: Registration Page 5

Fig 2.3: Task Page 6

Fig 2.4: Dashboard 7

Fig 2.5: Completed Task Page 7

Fig 2.6: Progress Task Page 8

Fig 2.7: Canceled Task Page 8

Fig 2.7: Profile Page 9

Fig 2.8: Mobile Views 9

Fig 3.1: Database model of the proposed project 11

Fig 3.4(a): User Data 13

Fig 3.4(b): Task Data 13

Fig 4.1: Flow chart 14

# Chapter 1



## Introduction

### Background

The Task Management System is a website-based project that can help a certain company to manage its project task progress or individual people to manage his daily task. This website is created by MERN (MongoDB, Express JS, React JS, Node JS). Many users can create account and create their task for different schedule as a reminder. User can manage his task, or profile. He can create a new task, and he can also update his task. If the task was completed, he can add his task as a completed category, He can also filter his task as a priority label, such as, urgent, later, avoid and so on, If he wants to know the task list as a priority basis, he need to just one click and then his priority basis task are filtered and visual in front to his screen.

### Objectives

* To make a website which manages one’s daily task.
* From any where or any device he can see his task, when he needs to do it.
* To make a user-friendly website interface for easier interaction with the users.

### 

### Platform

We have used ReactJS as a front-end and Express JS and NodeJS and MongoDB database as a backend.

**React JS**:

React is a JavaScript library for building user interfaces. When starting a React project, a [simple HTML page with script tags](https://reactjs.org/docs/add-react-to-a-website.html) might still be the best option. It only takes a minute to set up!

**NPM:**

NPM is the world's largest software registry. Open source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well.

**MongoDB:**

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

**Node JS:**

The API reference documentation provides detailed information about a function or object in Node.js. This documentation indicates what arguments a method accepts, the return value of that method. It also indicates which methods are available for different versions of Node.js.

**Express JS:**

### Many popular frameworks are based on Express. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

# Chapter 2



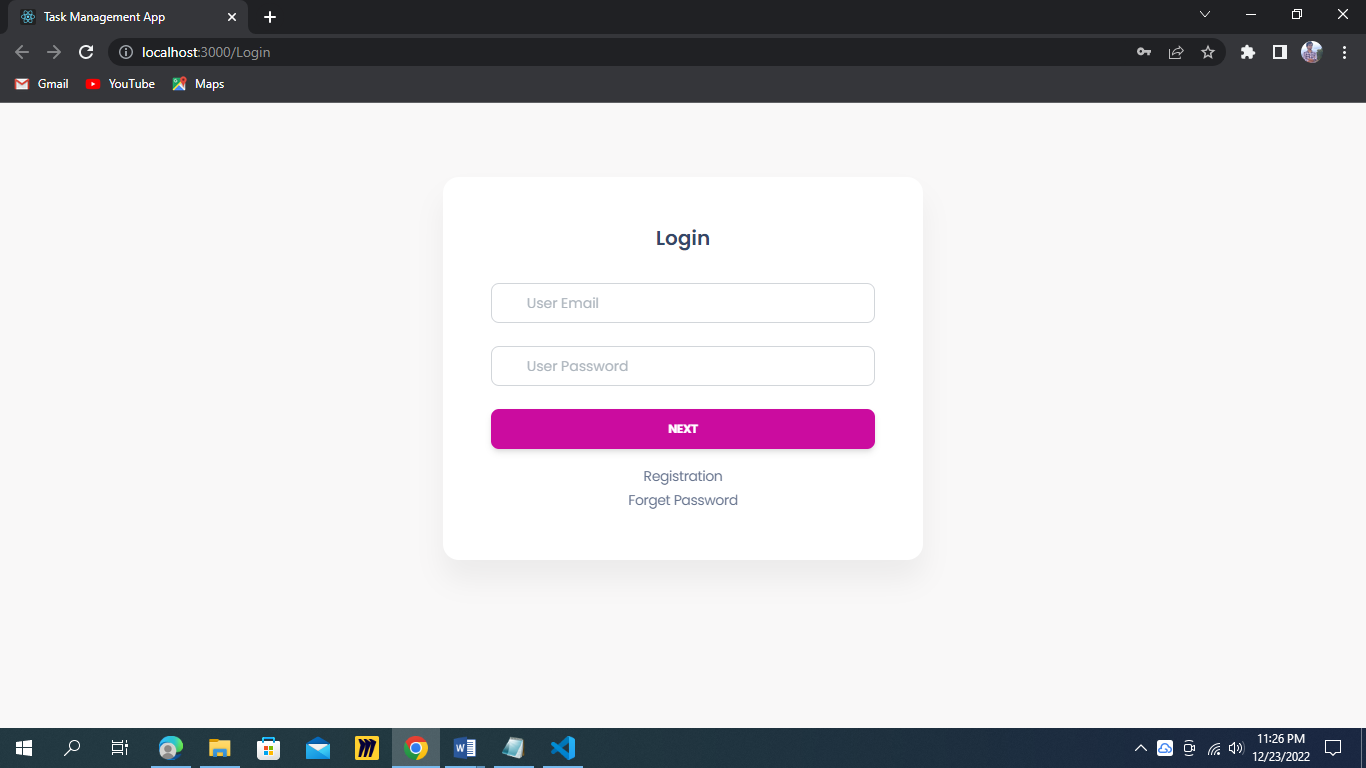
## The Interface

### 

### Login

This is the main interface for starting the whole system.

Everything from register to login, subscribe to projects creation, become a provide and accessing created projects can be done from the home interface.



### Fig 2.1: Login page

Here our login page, anyone can login from here, with his user email and password. If anyone don’t have an account than he just simply registers for his account.

### Register

Users have to register first to use the service. For registering, users need to provide some information like fist name, last name, email and a strong password and ﬁnally have to agree with our teams and conditions. Then clicking the register button, the ﬁrst step will be completed.

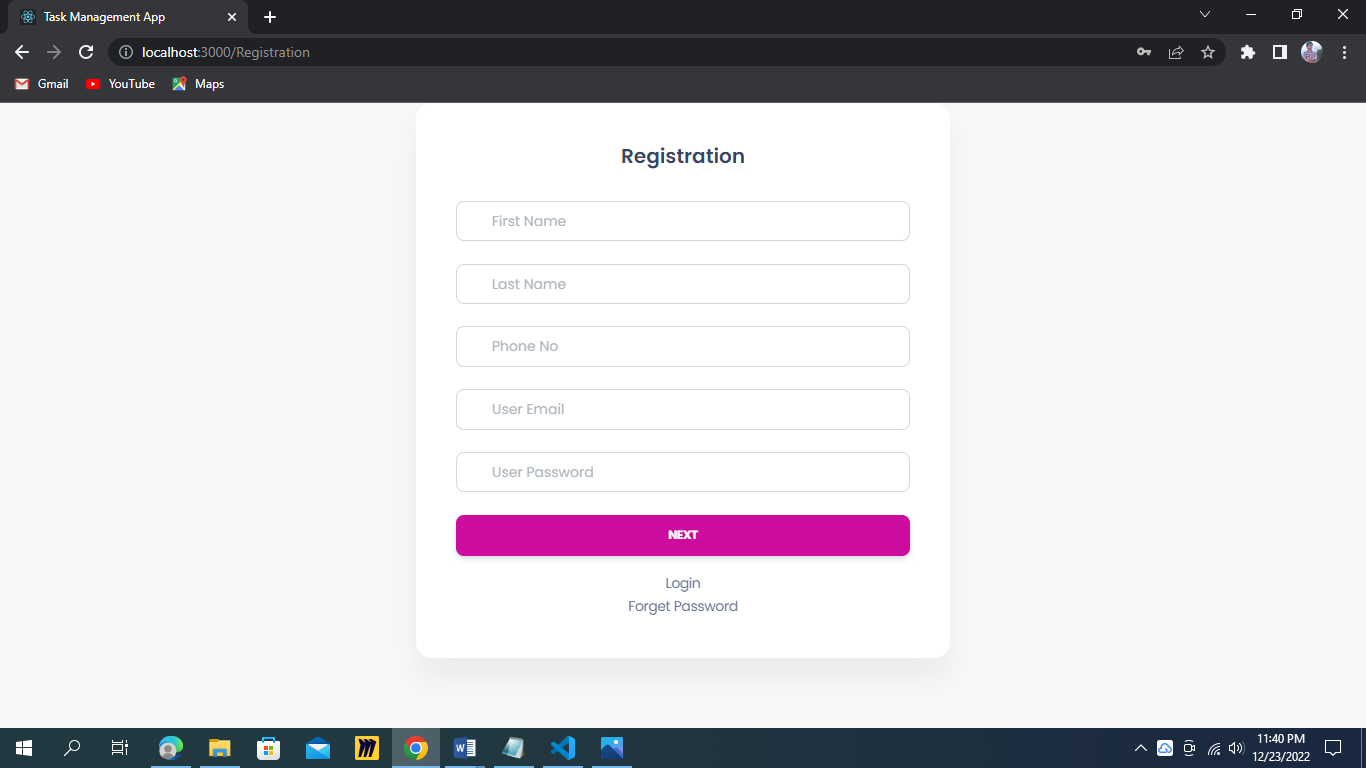


Fig 2.2: Registration Page

### Task

All the task which is scheduled are saved here.

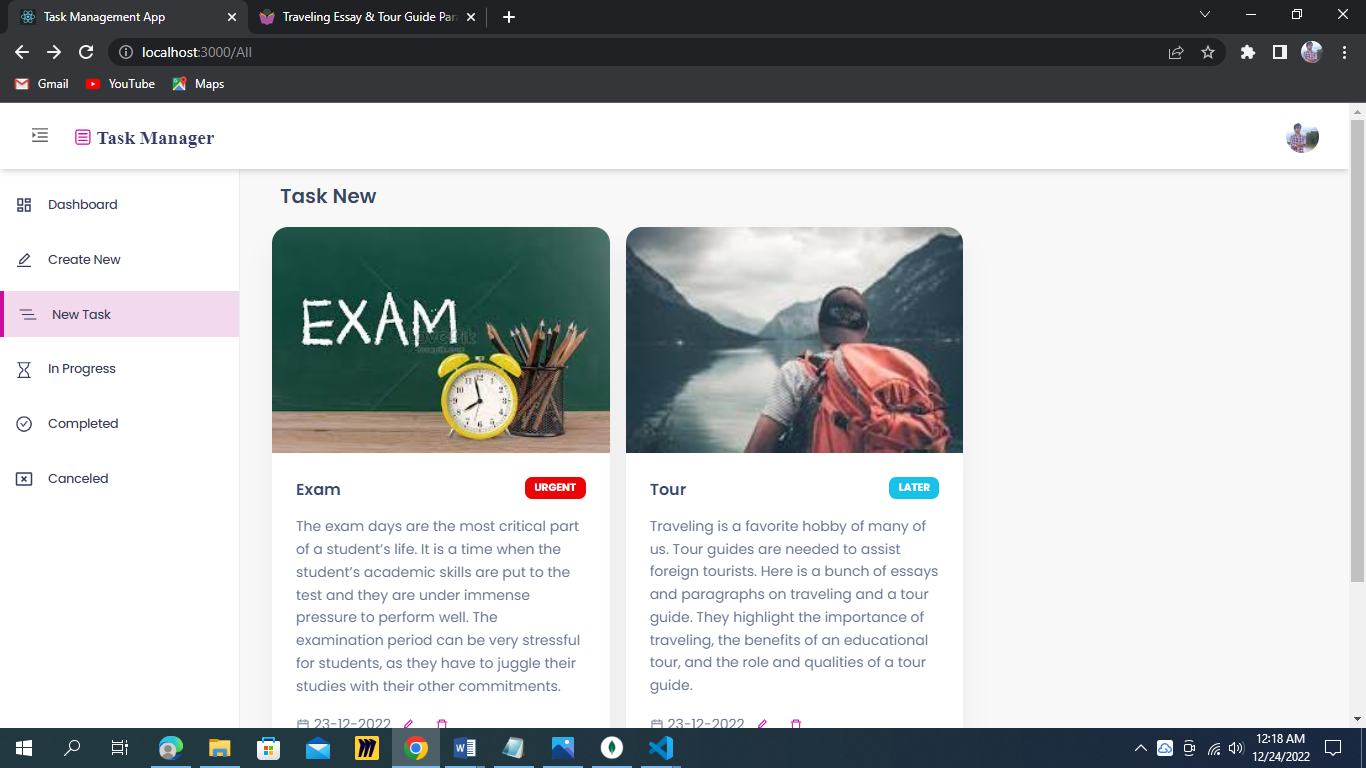


Fig 2.3: Task Page

**2.4 Dashboard:**

Here total task is shown.

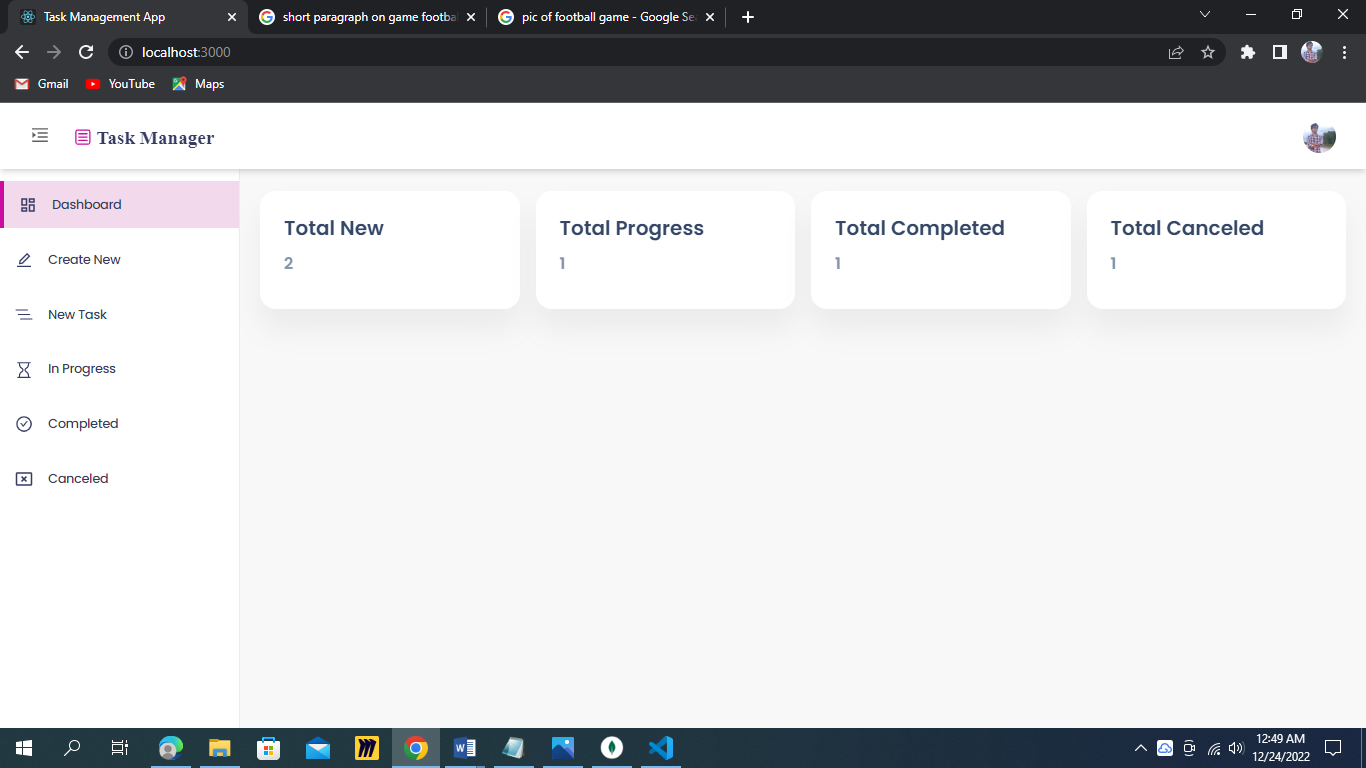


Fig 2.4: Dashboard

* 1. **Completed Task:**

Which was completed are stored in here.

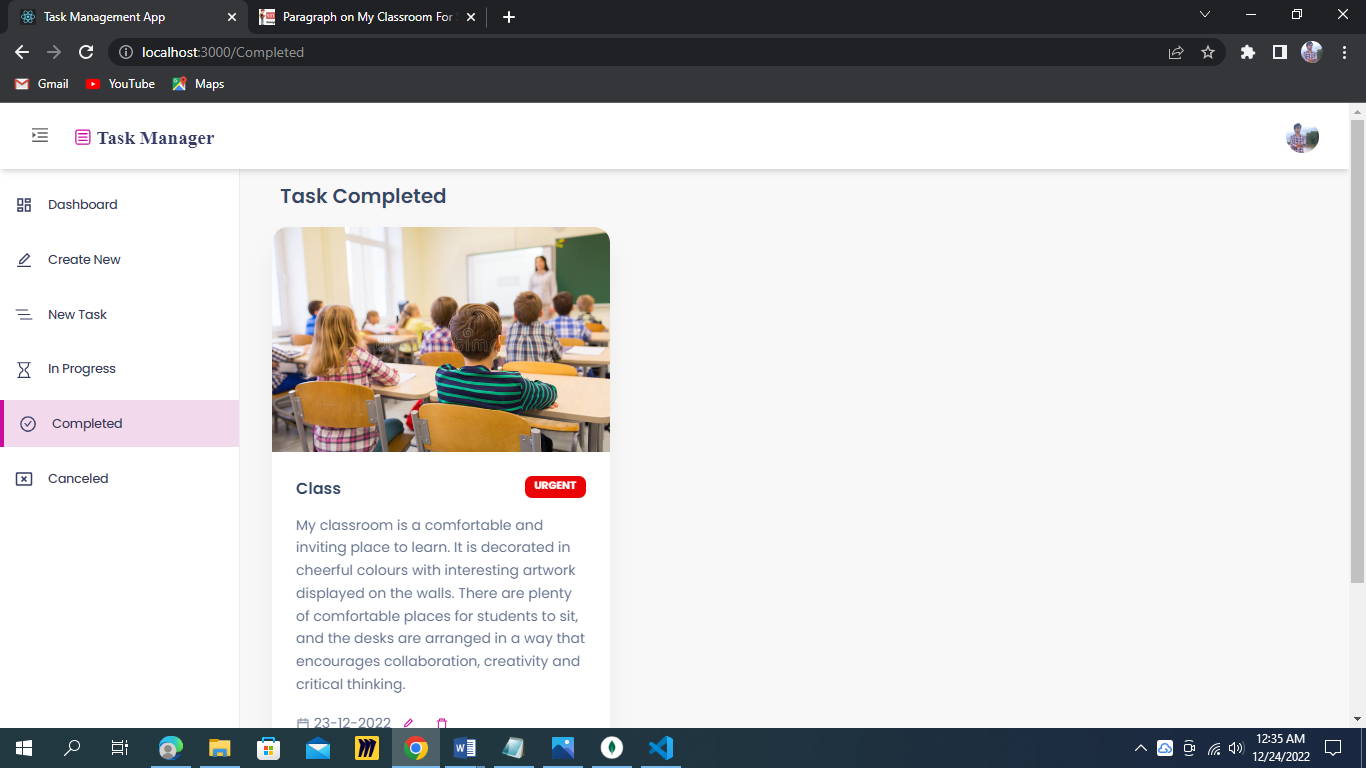


Fig 2.5: Completed Task Page

* 1. **Progress Task:**

Task which is in progress are stored in here. Or which we edited our task to progress are stored in here.

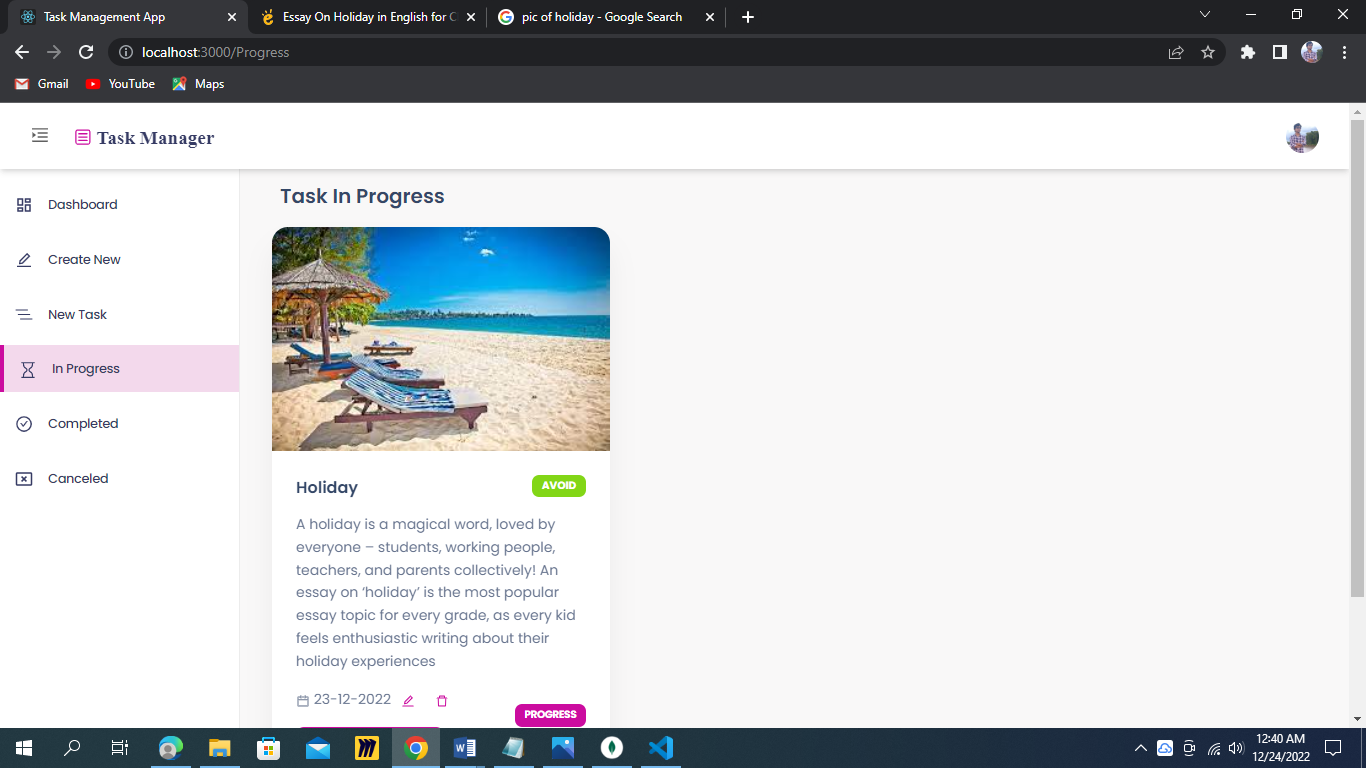


Fig 2.6: Progress Task Page

**2.7: Canceled Task:**

The task which task was canceled by us are stored here.

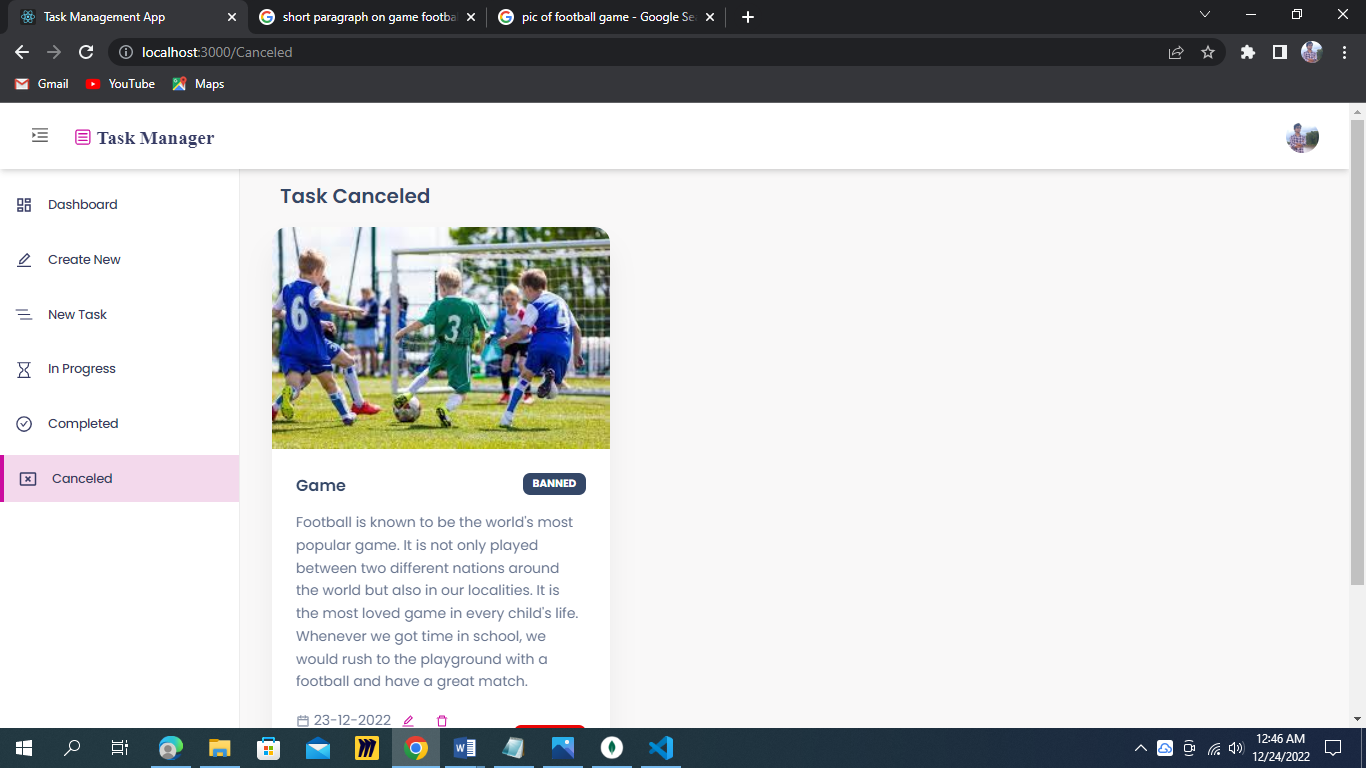


Fig 2.7: Canceled Task Page

**2.8: Profile:**

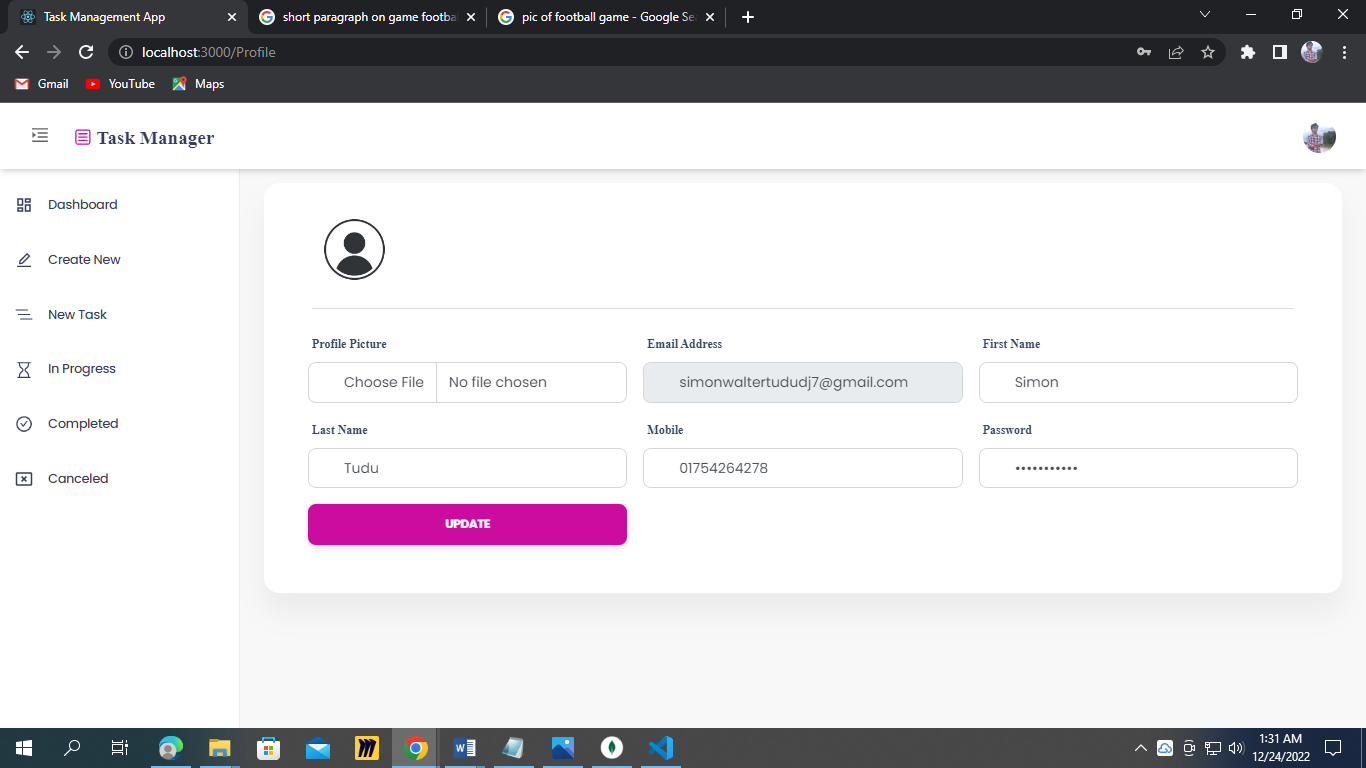
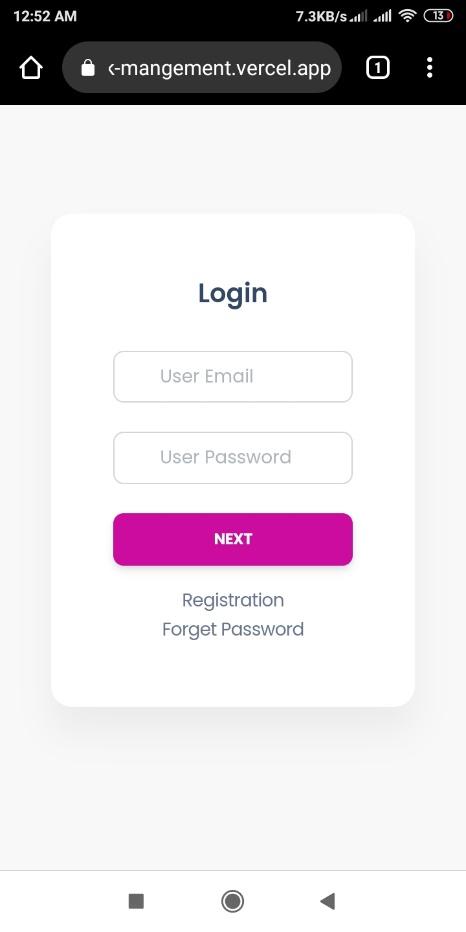
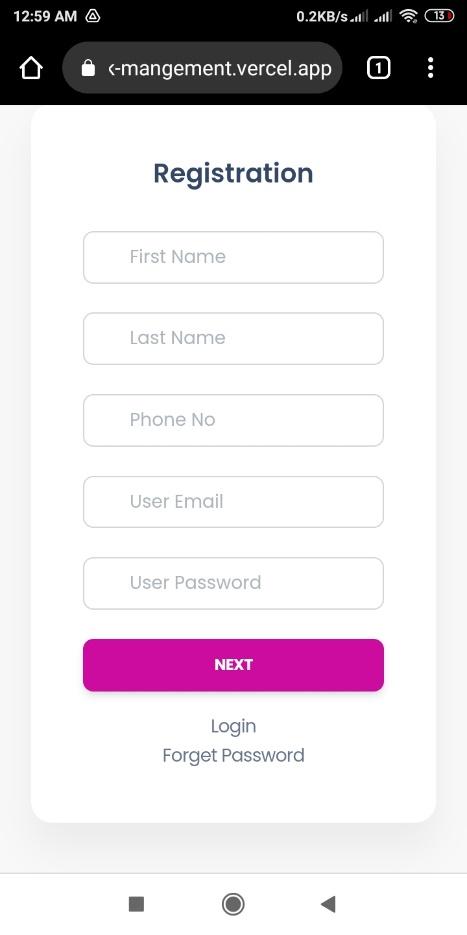
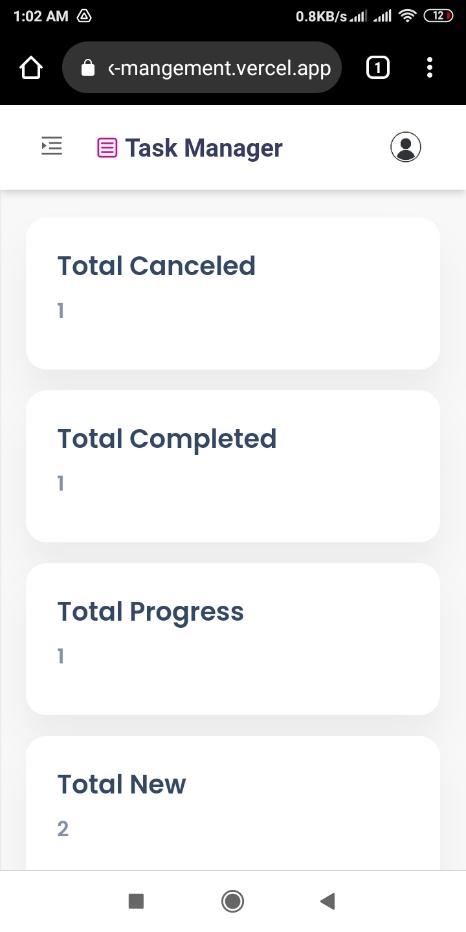
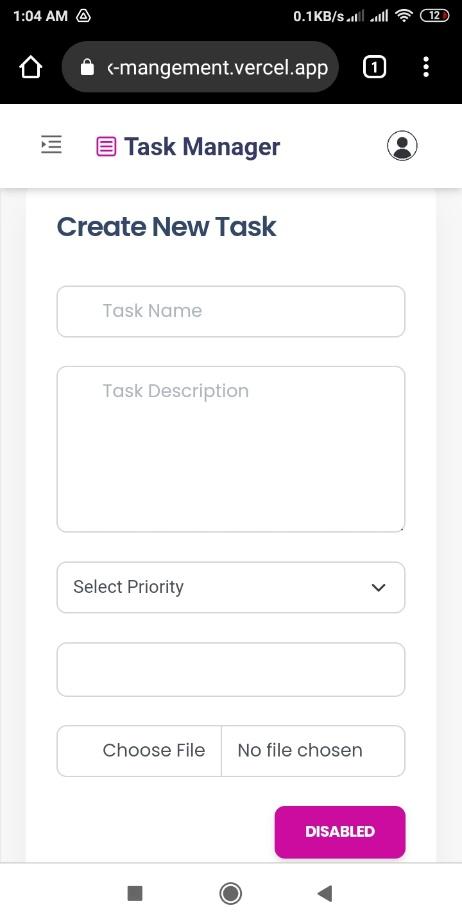
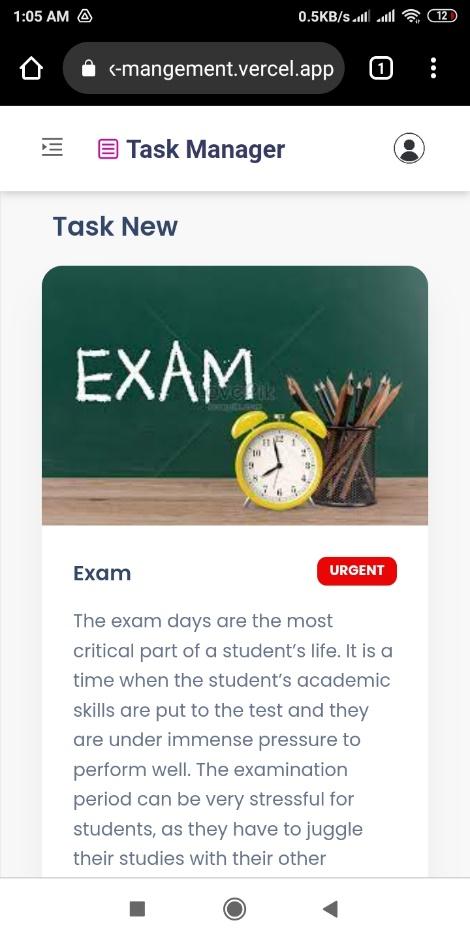
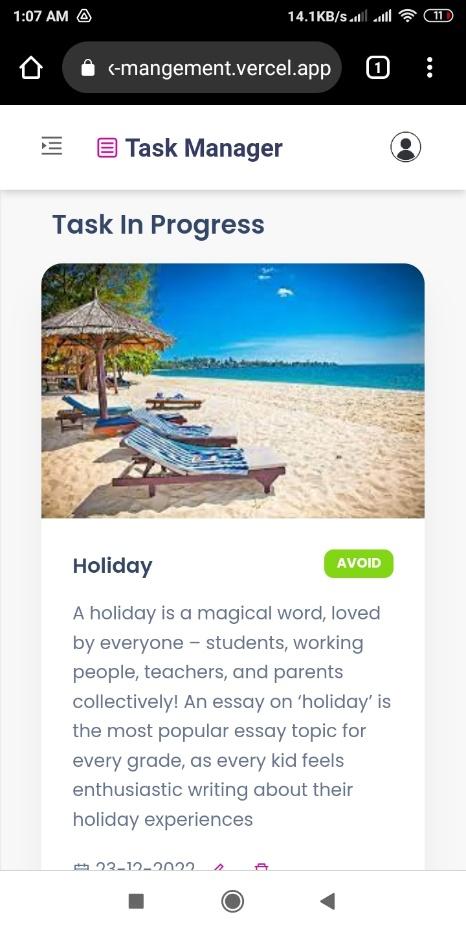
****

Fig 2.8: Profile Page

**2.9: Mobile Views:**

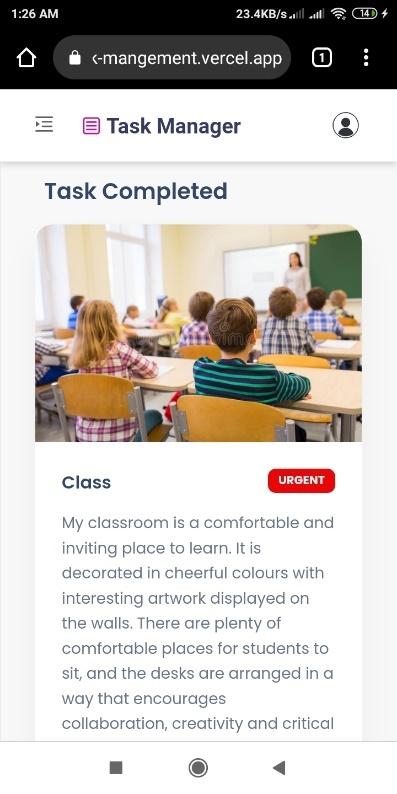
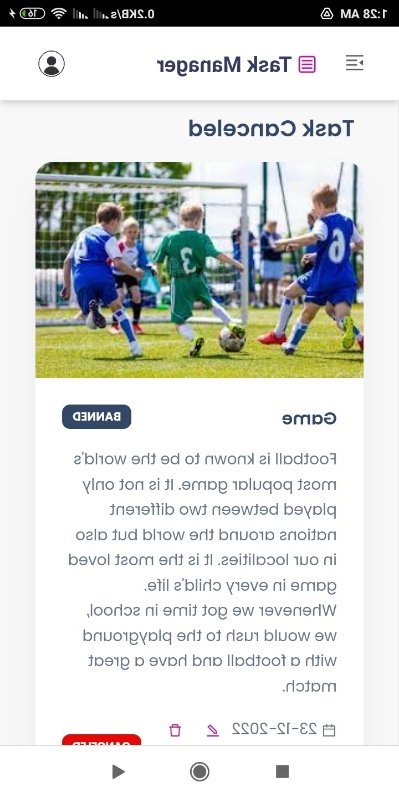
  

Fig 2.9: Mobile Views

# Chapter 3



**Methodology**

The proposed method is divided into several stages, including dataset collection, normalization, dataset splitting, and so on.

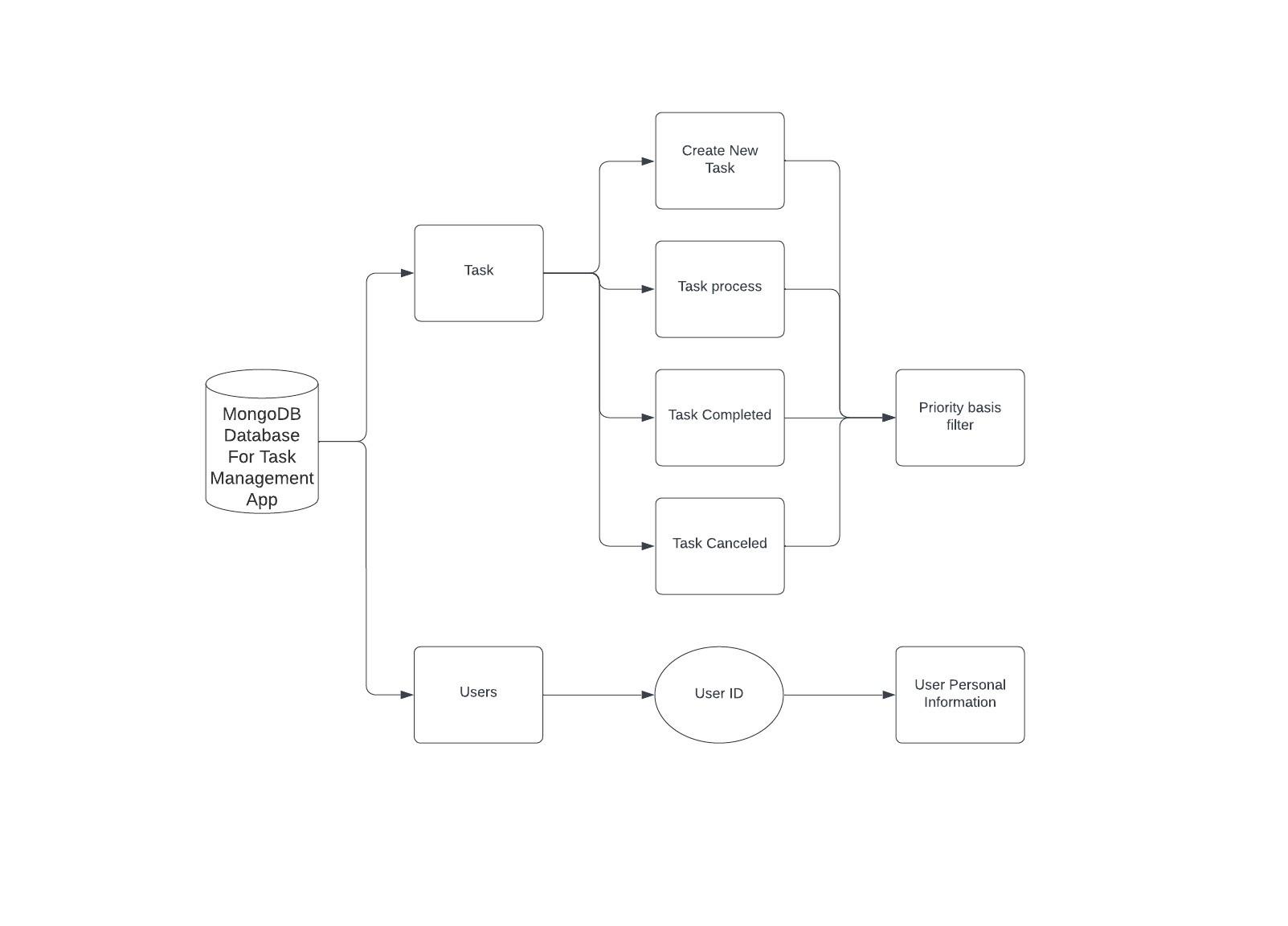


Figure 3.1: Database model of the proposed project

**3.1: Data Collection and Description:**

We collected data about users by manually registering through user’s in the website. We provided necessary information like user first name, last name, email address, mobile number, password and profile picture. It’s a website so, as possible as user can create an account. After successful registration, the user can edit their current information. For example, the user can only change password or profile picture etc. After registration he is allowed to create his personal task what he wants to add can add as more as he wishes. He can modify or edit his tasks in different category. For example, he wants to filter the tasks as he wants, or he can put that task in different category, or he can also add a specific picture, time schedule for a particular work.

3.2 MongoDB Database

We used MongoDB database, but the speed of the MongoDB Shell can decrease while processing massive data. Further, the MongoDB Shell works on the database on its own, and users cannot analyze and see the data changes independently. In contrast, in MongoDB Compass, users can visualize their database and its modifications without worrying about the performance.

This user id is also retrieved when he is logging in. So, whenever he is updating their fields, editing their task, going through the review option the authentic and actual data are shown.

3.3 Data manipulation

A user can manipulate data only for the following fields:

1. User ID and Password

2. Their personal “About” section

3. Their task schedule.

4. Priority basis task can be added.

They cannot change their full name, email, phone number, gender after registration.

The user can view the following their data or task from anywhere through mobile.

So It is very easy for anyone to access.

3.4: Data in MongoDB Views:



Fig 3.4(a): User Data

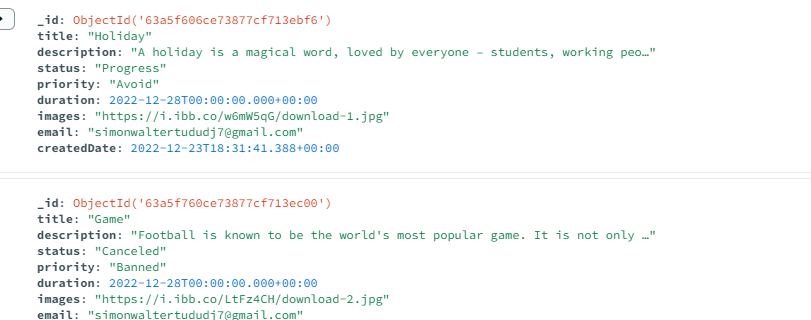


Fig 3.4(b): Task Data

# Chapter 4



**Flow Chart**

**4.1: Flow Chart**

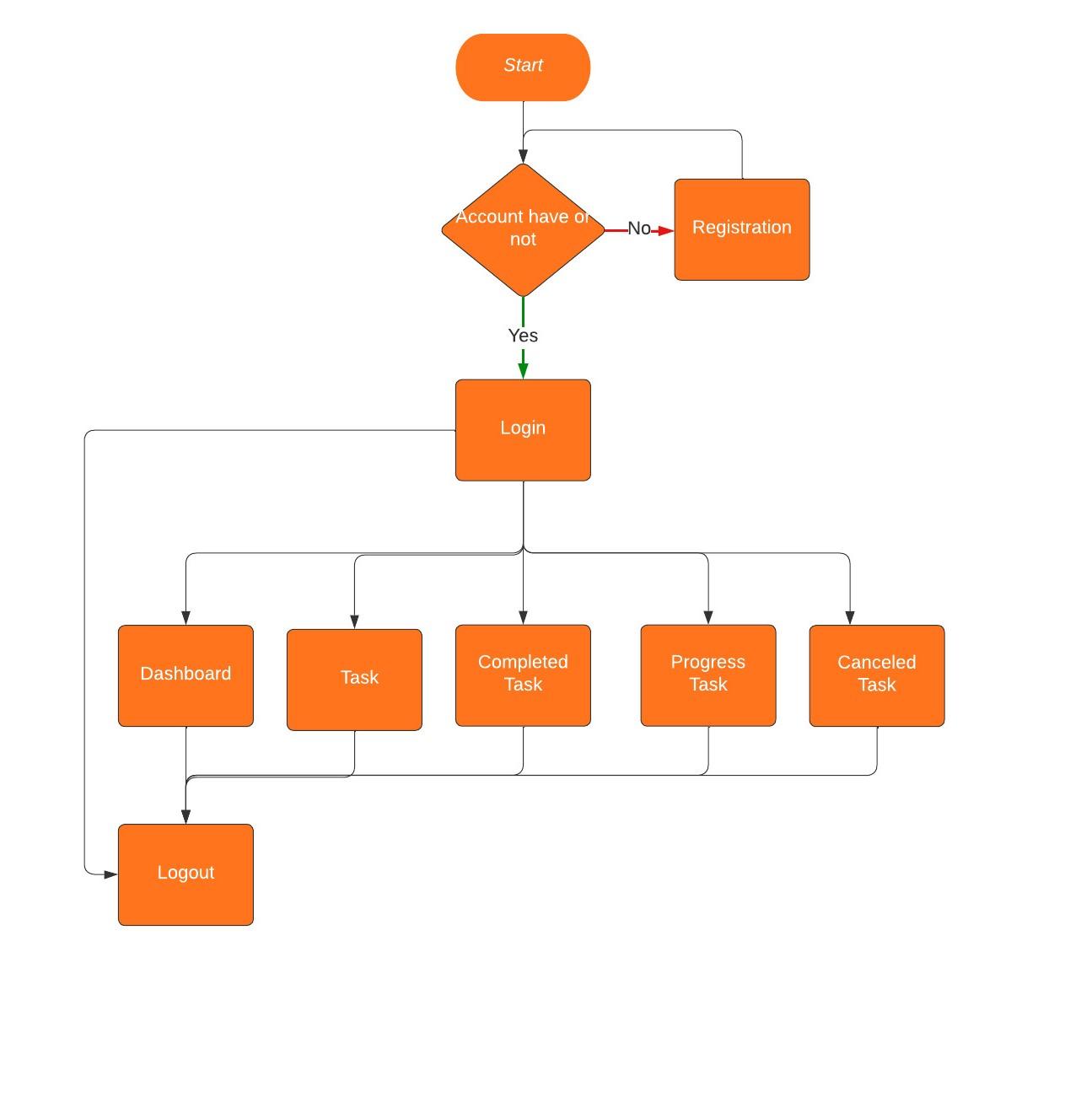


Fig 4.1: Flow Chart

# Chapter 5



## Limitations, Future Work, Conclusion

## 5.1: Limitations

* + 1. We can’t add forget password system.
    2. MongoDB database is slow.
    3. We can’t add OTP verification.

## 5.2: Future Work

The biggest future plane of our project is to make peoples working schedule as easy as possible. Anyone can get info about their task from anywhere through mobile handset.

## 5.3: Conclusion

We have learned many things throughout the project. Such as website build with MERN (MongoDB, ExpressJS, ReactJS, NodeJS), Database Management.

But we didn’t use everything we have learned. Rather those things are applied which are more compatible. Finally, we can say It was a great leaning experience.

# References



1. [Tutorial: Intro to React – React (reactjs.org)](https://reactjs.org/tutorial/tutorial.html)
2. [MongoDB Atlas Database | Multi-Cloud Database Service | MongoDB](https://www.mongodb.com/atlas/database)
3. [MongoDB Compass | MongoDB](https://www.mongodb.com/products/compass)
4. [Node.js (nodejs.org)](https://nodejs.org/en/)
5. [(475) Learn the MERN Stack - Full Tutorial (MongoDB, Express, React, Node.js) - YouTube](https://www.youtube.com/watch?v=7CqJlxBYj-M&ab_channel=freeCodeCamp.org)
6. [MongoDB Tutorial (tutorialspoint.com)](https://www.tutorialspoint.com/mongodb/index.htm)
7. <https://lucid.app/documents#/dashboard>
8. <https://vercel.com/templates>
9. [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)