

**A
Project III Report
On
Productivity Website**



**Submitted for Partial Fulfillment of the Award of Bachelor of Technology
(B.Tech) in CSE Kurukshetra University Kurukshetra**

Submitted By:

Paras Mayur 1220259

Prashant Wadhwa 1220266

Rahul Mittal 1220271

Submitted to: Ms. Simran (Assist. Professor)

Department of Computer Science and Engineering

**Seth Jai Parkash Mukand Lal Institute of Engineering & Technology
Affiliated to Kurukshetra University Kurukshetra**

Declaration

We hereby certify that the work which is being presented in the Project III Report entitled, “**Productivity Website**” by us, **Paras Mayur 1220259 , Prashant Wadhwa 1220266 and Rahul Mittal 1220271** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in **Computer Science Engineering** submitted in the Department of Computer Science and Engineering at JMIT Radaur (Affiliated to Kurukshetra University Kurukshetra, Haryana (India)) is an authentic record of our own work carried out under the supervision of **Ms. Simran**. The matter presented in the report has not been submitted in any other University/Institute for the award of any degree.

Paras Mayur, Prashant Wadhwa, Rahul Mittal

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Ms. Simran

Asstt. Professor Department of Computer Science and Engg. JMIT Radaur

Countersigned By: **Dr. Gaurav Sharma**

The B.Tech Project III Viva-voce examination of Candidate Paras Mayur (1220259), Prashant Wadhwa (1220266) and Rahul Mittal (1220271) was held on 23 May 2023 and was accepted.

Ms. Simran

(Supervisor)

(Examiner)

Acknowledgement

The writing of this project report has been assisted by the generous help of many people. We feel that we were very fortunate to receive assistance from them. We wish to express our sincere appreciation to them.

First and foremost, we are indebted to our principal supervisor, **Ms. Simran** (Asstt. Professor, Department of Computer Science and Engineering) of JMIT Radaur, who has been very supportive at every stage of our project completion. We wish to express our utmost gratitude to her for the invaluable advice and patience in reading, correcting and commenting on the drafts of this report and, more importantly, for his generosity which we have received throughout our project completion.

We would like to acknowledge and extend our heartfelt gratitude to Ms. **Simran** who helped and encouraged us throughout this journey. We wish to express our thanks to all staff members of JMIT Radaur, who also helped us in conducting this study. Finally, we are particularly indebted to our dearest parents/guardians as without their generous assistance and love; this project could never have been completed.

Table of Contents

Declaration	i
Acknowledgement	ii
Table of Contents	iii
List of Figures	iv
List of Tables	v

Chapter - 1	I II III IV V	Problem Background and Context System Objective Functionality Technical & Economic Feasibility Risk Factors and their Mitigation	1-4
Chapter - 2	I II	Primary Research Techniques and Analysis Secondary Research	5-7
Chapter-3	I II	Requirement Specification Design Analysis	8-13
Chapter-4		System Implementation	14-16
Chapter-5		Project Details	17-22
Chapter - 6		Future Scopes and Limitations	23
Chapter-7		Test Cases	24-25
Chapter - 8		Conclusion	26
Chapter - 9		Bibliography	27
Chapter-10		Appendix	28

List of Figures

Figure	Description	Page Number
Figure 1	This image shows the signup page. Users can signup and login to the dashboard.	
Figure 2	The user can login to the dashboard.	
Figure 3	It shows the dashboard of the user, currently with 1 project DSK.	
Figure 4	This image shows the tasks with categories within a single project DSK.	
Figure 5	This image shows the newly added projects college and home in the dashboard.	
Figure 6	This image shows the updated dashboard with one project removed from the list.	

Chapter 1

Problem Background and Context

System Objective

Functionality

Technical & Economic Feasibility

Risk Factors and their Mitigation

I Problem background and Context

The productivity app builds a suite of analytical tools for productivity to accelerate growth across your lifecycle.

Why did I create a Productivity app?

A productivity app can be incredibly beneficial in today's fast-paced and multitasking world. Here are several reasons why we must consider using a productivity app:

- **Task Management:** A productivity app helps you stay organized by allowing you to create and manage tasks efficiently. You can set priorities, deadlines, and reminders, ensuring that nothing falls through the cracks.
- **Goal Setting:** Productivity apps often include goal-setting features that allow you to define your objectives and track your progress. By setting clear goals and monitoring your achievements, you can stay motivated and increase your overall productivity.
- **Collaboration and Teamwork:** If you work in a team or collaborate with others, productivity apps can facilitate communication, task delegation, and project management. They often provide shared calendars, file sharing, and real-time collaboration features to streamline teamwork and enhance productivity.

Introducing our Productivity app, the ultimate tool for efficient task management, time optimization, and goal achievement. With a user-friendly interface, our app helps you organize and prioritize tasks, set deadlines, and receive reminders, ensuring that nothing falls through the cracks. Stay focused and eliminate distractions using our built-in timers and productivity techniques.

Collaborate seamlessly with team members, delegate tasks, and track projects in real time. Capture ideas, create to-do lists, and store information in one centralized location. Gain valuable

insights into your productivity patterns and make data-driven decisions for improvement. Our app integrates with other tools and platforms, keeping your workflow streamlined and synchronized. Boost your productivity and achieve your goals with our powerful productivity app.

II System Objective

- **Simple UI:** Our productivity app features a clean and intuitive user interface, making it easy to navigate and use. With a minimalist design and streamlined functionality, you can quickly access all the features and tools without any unnecessary complexity.
 -
- **Team Collaboration:** Our app fosters seamless collaboration among team members, providing shared calendars, task delegation, and real-time project management. Collaborate efficiently, assign tasks, and track progress together, ensuring that everyone stays on the same page and achieves collective goals.
 -
- **Easy Drag and Drop:** Our app simplifies task management with a user-friendly drag and drop feature. Effortlessly prioritize and rearrange tasks with a simple gesture, streamlining your workflow and making organization a breeze.

III Functionality

- People can create different lists according to their tasks.
- They can create different categories within the list.
- They can create as many as tasks inside the category.
- They can shift the tasks from one category to another.
- Also, they can collaborate with other people on the same list.

IV Technical and Economic Feasibility

Feasibility Study:

A feasibility study is an analysis that takes all of a project's relevant factors into account including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need to significantly alter the original goal.

Technical Feasibility:

A technical feasibility study assesses the details of how you intend to deliver a product or service to customers. Think materials, labor, transportation, where your business will be located, and the technology that will be necessary to bring all this together.

Productivity Platform rates high on technical feasibility as it can be developed using existing tools and frameworks and does not require the development of new tools or frameworks. The system does not have a complex architecture and the approach adopted is tried and tested.

Economic Feasibility:

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its

strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success. It consists of market analysis, economic analysis, technical and strategic analysis.

Productivity Platform rates low on the economic feasibility rating. While the infrastructure required to support the project is not expensive, the return on investment might not be satisfactory. Few people will be willing to pay for email services. However, this is still not a deal-breaker as the cost of going to market is very less in itself.

V Risk Factor and Mitigation

Business risk is the exposure a company or organization has to factor(s) that will lower its profits or lead it to fail. Anything that threatens a company's ability to meet its target or achieve its financial goals is called business risk. These risks come from a variety of sources, so it's not always the company head or a manager who's to blame. Instead, the risks may come from other sources within the firm or they may be external from regulations to the overall economy.

While a company may not be able to shelter itself from risk completely, there are ways it can help protect itself from the effects of business risk, primarily by adopting a risk management strategy.

One of the major risk factors associated with Web Analytics Platform is that very few people might be willing to pay for a service despite the extra features.

Due to this, the overall infrastructure cost should be kept at a minimum and the system should be designed such that alternative economic models such as advertising can be incorporated later on. Also, developing an MVP before going all-in should be considered.

Chapter 2

Primary Research Techniques and Analysis

Secondary Research

I Primary Research Techniques and Analysis

Primary research techniques and analysis refer to the methods used to collect original data directly from the source, such as users or target audience, to gain insights and inform decision-making. It involves gathering firsthand information through various means and analyzing the collected data to draw meaningful conclusions. Some common primary research techniques include surveys, interviews, observations, experiments, and user testing. The data collected through these methods is then analyzed using qualitative or quantitative analysis techniques, depending on the nature of the data. Qualitative analysis involves interpreting and categorizing textual or visual data, while quantitative analysis involves statistical analysis to identify patterns, correlations, and trends. The findings from primary research techniques and analysis provide valuable insights into user needs, preferences, behaviors, and attitudes, enabling businesses to make informed decisions and improve their products or services.

When conducting primary research techniques and analysis in a React project, there are several approaches to consider:

- 1. Surveys and Questionnaires:** Implement survey forms within your React application to collect data and feedback from users. You can use React's form capabilities to create interactive and responsive survey forms, validate input, and capture responses. Analyze the collected data to gain insights and inform decision-making.
- 2. User Interviews:** Conduct interviews with users to gather qualitative data about their experiences, preferences, and pain points. You can integrate audio or video communication features into your React application for conducting remote interviews. Analyze the interview transcripts for common themes, patterns, and user insights.
- 3. User Testing and Feedback:** Incorporate user testing sessions within your React application to observe how users interact with the interface and gather their feedback. React's

testing utilities, such as Jasmine and Protractor, can assist in creating automated tests and capturing user actions. Analyze the test results and user feedback to identify usability issues and areas for improvement.

4. Analytics Integration: Integrate analytics tools, such as Google Analytics or Mixpanel, into your React application to gather quantitative data on user behavior, page views, conversions, and more. Utilize React's routing and lifecycle hooks to track user actions and events. Analyze the collected data to gain insights into user engagement, identify bottlenecks, and make data-driven decisions.

When analyzing the data collected from these techniques, we consider using data visualization tools like D3.js or React Material's charts to present findings in a visually appealing manner. This will help stakeholders and development teams understand the research insights more effectively and make informed decisions for enhancing the React project.

II Secondary research

Secondary research refers to the process of gathering and analyzing existing information and data that has been previously collected by others. In an React project, secondary research can be valuable for gaining insights, understanding industry trends, and informing decision-making. Here are some common techniques and sources for conducting secondary research in an React project:

1. Literature Reviews: Review academic journals, research papers, books, and articles related to web development, React, and relevant topics. This helps you understand best practices, emerging trends, and established techniques in React development.

2. Online Documentation and Official Guides: Explore React's official documentation, including guides, tutorials, and API references. These resources provide in-depth information about React's features, components, modules, and best practices.

3. Online Forums and Communities: Engage with online communities, such as React's official forum, Stack Overflow, or Reddit's React community. These platforms offer a wealth of information, discussions, and problem-solving insights shared by experienced React developers.

4. Case Studies and Success Stories: Study case studies and success stories of real-world React projects. These can be found in online publications, developer blogs, and tech company websites. Analyzing successful implementations and their challenges can provide valuable insights and inspiration for your project.

5. Industry Reports and Surveys: Access industry reports, surveys, and market research conducted by organizations and research firms. These reports often provide statistics, trends, and insights about web development practices, including the adoption and usage of React.

6. Open-Source Projects and Libraries: Explore open-source React projects and libraries available on platforms like GitHub. Analyzing the code, documentation, and discussions around these projects can provide valuable insights and reusable components for your development.

7. Technology Blogs and News Websites: Follow technology blogs and news websites that cover React and web development-related topics. These sources often publish articles, tutorials, and updates on the latest trends, best practices, and new features in React.

Chapter 3

Requirement specifications

Design Analysis

I Requirement specifications

Skill Set Required

ReactJS is an open-source JavaScript library that is used for building user interfaces in a declarative and efficient way. It is a component-based front-end library responsible only for the view layer of an MVC (Model View Controller) architecture. React is used to create modular user interfaces and it promotes the development of reusable UI components that display dynamic data.

Software Requirement Spec

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Purpose of SRS

- To implement the system according to customer needs
- To ensure the software developers that the product being developed by them is exactly what is required by the customer.
- To help the maintenance engineers to understand the functionality of the system.
- To aid the user document writers to write the user manuals correctly.

Functional Requirements

User Interface

- The application must be user friendly and intuitive. It must prompt the user with proper message boxes wherever required.
- To help them perform various actions and how to proceed further, the system must respond normally under any in/out conditions and display proper results instead of throwing faults and errors.

Non Functional Requirements

- **Performance Requirements**

→ A decent system is required for the proper functioning of this application.

- **Safety Requirements**

→ No harm to anyone using this application.

- **Product Security Requirements**

→ The product is secured, basically physical security is not the issue in this kind of application.

- **Data Security**

→ As this application allows users to create messages and save them, extensive measures should be taken to ensure data safety.

What to expect?

Here are some of the features available:

- **Sign In**

The application allows users to "Sign In" when they have already signed up. Else lets them sign up first.

- **Dashboard/Profile page**

After logging in, the user is redirected to their Dashboard page. From here, they can monitor all their projects, and members, and delete and add more projects.

- **Add members**

The app lets you collaborate with your team members to manage the tasks efficiently and be more productive.

Hardware Requirements

Any modern laptop or mobile phone with an up-to-date browser and internet connection may use this application.

II Design Analysis

Design is the first step in the development phase for any engineered product or system. Design is a creative process. A good design is the key to an effective system. The term "design" is defined as "the process of applying various techniques and principles to define a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles to define a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software

engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document for the programmers or database personnel.

System design goes through two phases of development:

- Logical Design
- Physical Design

Logical Design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system - its data flows, file content, volumes, frequencies etc.
- Prepares output specifications that is, determines the format, content and frequency of reports.
- Prepares input specifications format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

Physical Design

Physical system produces the working systems by defining the design specifications that tell the programmers exactly what the candidate system must do.

It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flows through the system and a physical design walkthrough.
- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits, costs, and conversion date and system constraints.

Input Design

Input design is part of the overall system design that requires special attention; designing input data is to make the data entered easily and free from errors. The input forms are designed using the controls available in the UI kit CSS framework. Validation is made for every data that is entered. Help information is provided so that the users do not face any difficulties in operating the software.

Input design is the process of converting user-originated inputs to a computer-based format. A system user interacting through a workstation must be able to tell the system whether to accept the input to produce reports. The collection of input data is considered to be the most expensive part of the system design. Since the input has to be planned in such a manner so that relevant information is returned, extreme care is taken.

Output Design

Output design this application generally refers to the ease of scheduling emails through the web frontend using the given features.

- The output is designed in such a way that it is attractive, convenient and informative.
- Forms are designed with various features, which make the on-submit output more easy.
- As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision making. Form design elaborates the way output is presented and the layout available for capturing information.
- One of the most important factors of the system is the output it produces. This system refers to the results and information generated. Basically the output from a computer system is used to communicate the result of processing to the user.

Chapter 4

System Implementation

System Implementation uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases. These system elements are then integrated to form intermediate aggregates and finally the complete system-of-interest (Sol).

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). System elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing, the software realization processes of coding and testing, or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

—Backend

—JWT

—jwt.js

—Models

—model.js

—project.js

—node_modules

—index.js

—package.json

—package-lock.json

—Frontend

—node_modules

—**.Javascript**

—**src**

—**app**

—**defaultData**

—categories.ts

—data.ts

—**Guard**

—auth.guard.spec.ts

—auth.guard.ts

—**Modules/admin**

—components

—admin-dashboard

—contact

—dashboard

—footer

—header

—home

—admin-routing.module.ts

—admin.module.ts

—**components**

—forgot-password

—login

—not-found

—register

—**services**

—auth.service.ts

—user-details.service.ts

—**app-routing.module.ts**

—**app.component.html**

- app.component.scss
 - app.component.spec.ts
 - app.component.ts
 - App.module.ts
- assets
 - gitkeep
 - Photo.jpg
- environments
 - Environment.prod.ts
 - environment.ts
- favicon.ico
- index.html
- main.ts
- polyfills.ts
- styles.scss
- Test.ts
- browserslistrc
- editorconfig
- gitignore
- README.md
- Javascript.json
- karma.conf.js
- package-lock.json
- package.json
- tsconfig.app.json
- tsconfig.json

Chapter 5

Project Details

Project Details

In this section, we will provide an overview of our project, highlighting the technology used, key features, and a description of how the project works. This comprehensive overview will give you a clear understanding of the scope and purpose of our project, as well as the benefits it offers to users. Let's dive in and explore the exciting details of our endeavor

The project uses a combination of web technologies to create a dynamic and interactive website. The technologies used are:

HTML: A markup language that defines the structure and content of web pages. HTML uses tags to mark up different elements, such as headings, paragraphs, images, links, forms, etc. HTML also allows embedding other types of content, such as CSS, JavaScript, SVG, etc. In the project, HTML is used to create the basic structure and content of the web pages, such as the header, footer, navigation, main section, etc.

- **CSS:** A style sheet language that specifies the appearance and layout of web pages. CSS uses rules to apply style and formatting to HTML elements, such as colors, fonts, margins, padding, borders, backgrounds, etc. CSS also supports features like media queries, transitions, animations, flexbox, grid, etc. In the project, CSS is used to add style and formatting to the HTML elements, such as making them responsive, aligning them properly, adding hover effects, etc.
- **JavaScript:** A scripting language that enables interactivity and functionality on web pages. JavaScript can manipulate the HTML elements and their attributes, handle user input and events, fetch data from external sources, perform calculations and logic operations, etc. JavaScript also supports features like objects, arrays, functions, promises, async/await, etc. In the project, JavaScript is used to add interactivity and functionality to the web pages, such as validating forms, fetching data from APIs, updating the UI based on user actions or data changes, etc.

- **React.js:** A JavaScript library for building user interfaces using reusable components and state management. React.js uses a declarative syntax called JSX that allows writing HTML-like code in JavaScript. React.js also uses a concept called props to pass data and events between components. React.js also supports features like hooks, custom hooks, context API, etc. In the project, React.js is used to simplify the development of complex user interfaces by using components that can have their own state and props. For example, a component for a card that displays an image and some text can be reused multiple times with different data passed as props.
- **Figma:** A cloud-based design and prototyping tool that allows creating and collaborating on web and mobile app designs. Figma uses vector graphics to create UI elements and layouts that can be scaled and edited easily. Figma also supports features like grids, typography, colors, icons, images, etc. Figma also allows creating interactive prototypes that can simulate user interactions and transitions between screens. In the project, Figma is used to design the UI components and layouts using vector graphics, grids, typography, colors, icons, images, etc. Figma also allows creating interactive prototypes that can simulate user interactions and transitions between screens.
- **Tailwind CSS:** A utility-first CSS framework that provides low-level classes for styling web elements without writing custom CSS. Tailwind CSS follows a consistent naming convention and scale for its classes that makes it easy to use and remember. Tailwind CSS also supports features like responsive design, dark mode, customization, plugins, etc. In the project, Tailwind CSS is used to style the UI components using predefined classes that follow a consistent naming convention and scale. Tailwind CSS also helps achieve responsive design, dark mode, customization, etc.

These technologies work together to create a modern and responsive website that meets the project requirements. HTML defines the basic structure and content of the web pages, such as headings, paragraphs, images, links, etc. CSS adds style and formatting to the HTML

elements, such as colors, fonts, margins, padding, etc. JavaScript adds interactivity and logic to the web pages, such as handling user input, fetching data, manipulating the DOM, etc. React.js simplifies the development of complex user interfaces by using components, which are reusable pieces of UI code that can have their own state and props. Figma helps design the UI components and layouts using vector graphics, grids, typography, colors, etc. Tailwind CSS helps style the UI components using predefined classes that follow a consistent naming convention and scale.

Features:

Todo list feature:

Enables users to create and organize tasks into different lists and categories.

Allows users to set due dates and priorities for their tasks.

Enables users to mark tasks as completed or incomplete.

Helps users to keep track of their progress and achievements.

Calendar feature:

Enables users to view their tasks and events on a monthly, weekly, or daily basis.

Allows users to add, edit, or delete tasks and events on the calendar.

Enables users to sync with other calendars such as Google Calendar or Outlook.

Helps users to plan ahead and avoid conflicts or overlaps.

Clock feature:

Enables users to see the current time and date on the website.

Allows users to set alarms for important events or deadlines.

Enables users to customize the sound and volume of the alarms.

Helps users to stay on schedule and be punctual.

Reminder feature:

Enables users to set notifications for their tasks and events.

Allows users to choose the date and time or the frequency of the notifications.

Enables users to customize the message and sound of the notifications.

Helps users to remember their tasks and events and avoid missing them.

Stopwatch feature:

Enables users to measure the duration of their tasks and activities.

Allows users to start, stop, pause, or reset the stopwatch.

Enables users to use the Pomodoro technique to improve their focus and productivity.

Helps users to manage their time and energy efficiently and effectively.

Notes feature:

Enables users to create and edit notes with rich text formatting, images, links, etc.

Allows users to attach notes to their tasks or events.

Enables users to search, sort, or filter notes by keywords, dates, categories, etc.

Helps users to capture and organize their ideas, information, feedback, etc.

News feature:

The news section feature is a curated feed of relevant and interesting news articles from various sources that users can browse and read on the site.

The news section feature helps users stay informed and updated on topics that matter to them, such as business, technology, health, entertainment, etc.

By incorporating these features into our productivity app, we aim to provide users with a versatile and collaborative platform that empowers them to stay organized, manage tasks effectively, and collaborate seamlessly with other.

Images of our Working Project

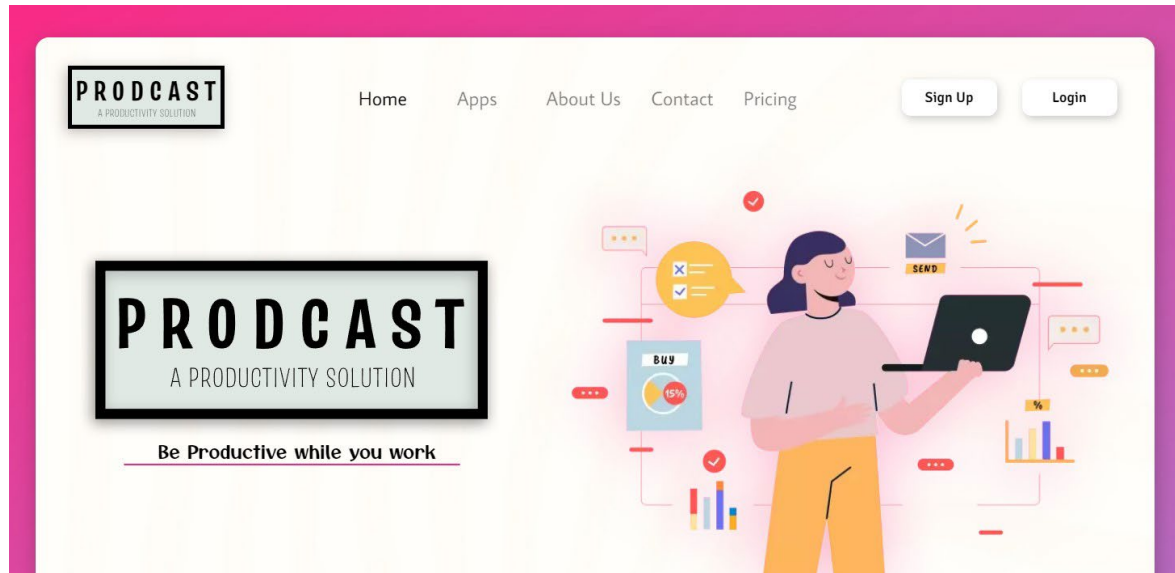


Figure 1

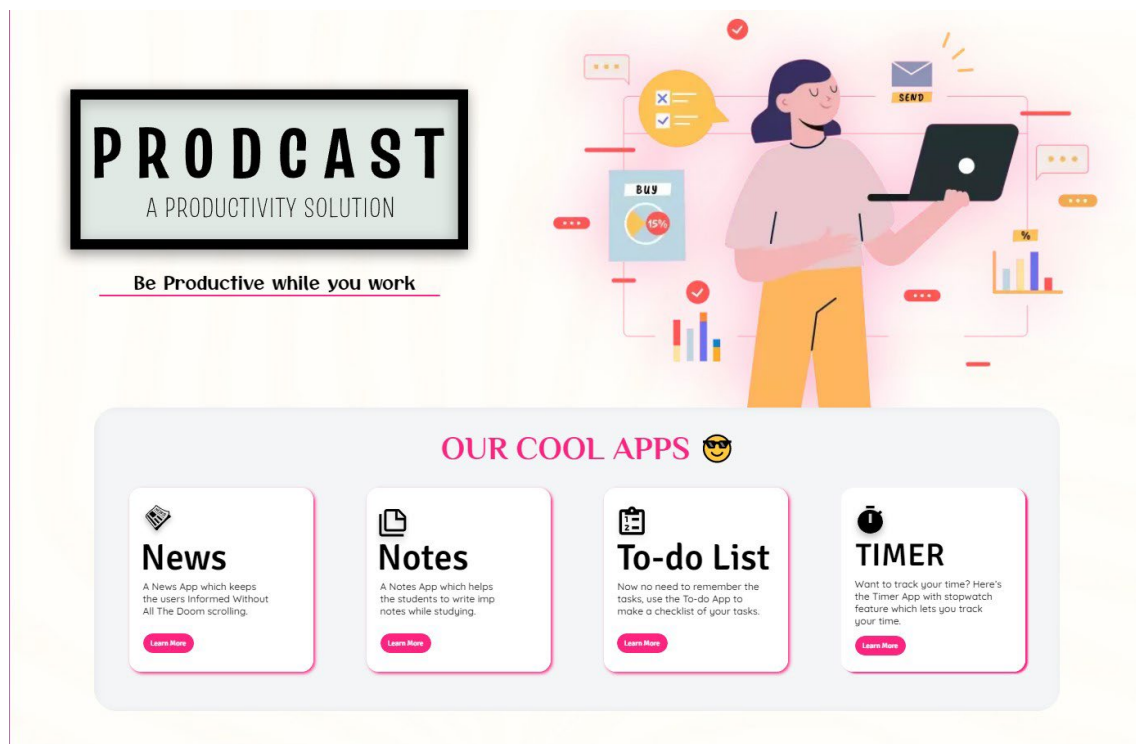


Figure 2

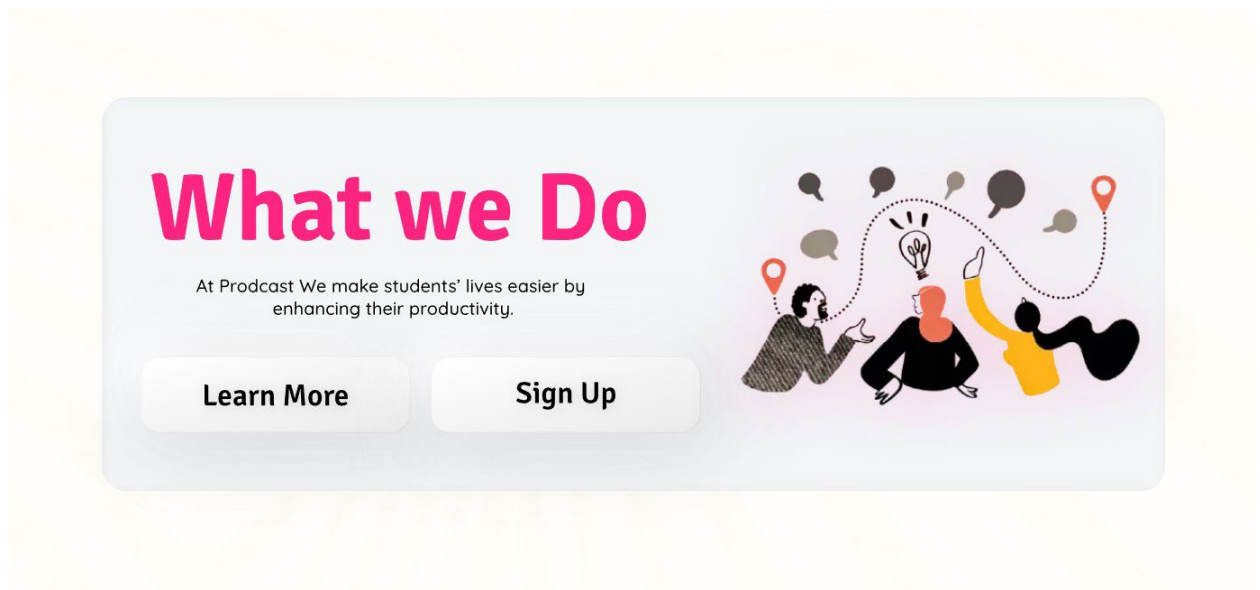


Figure 3

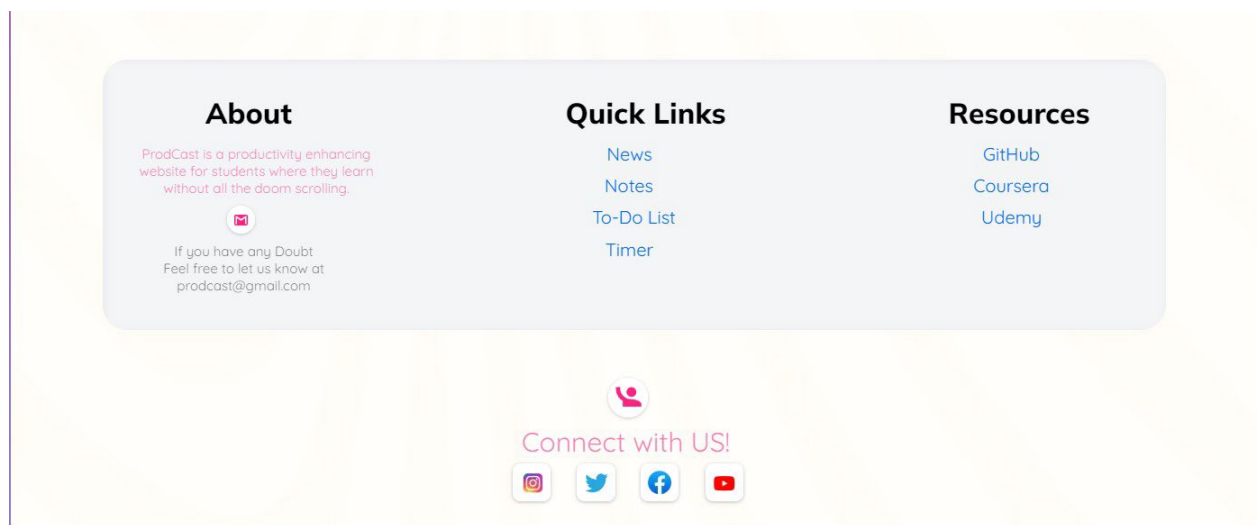


Figure 4

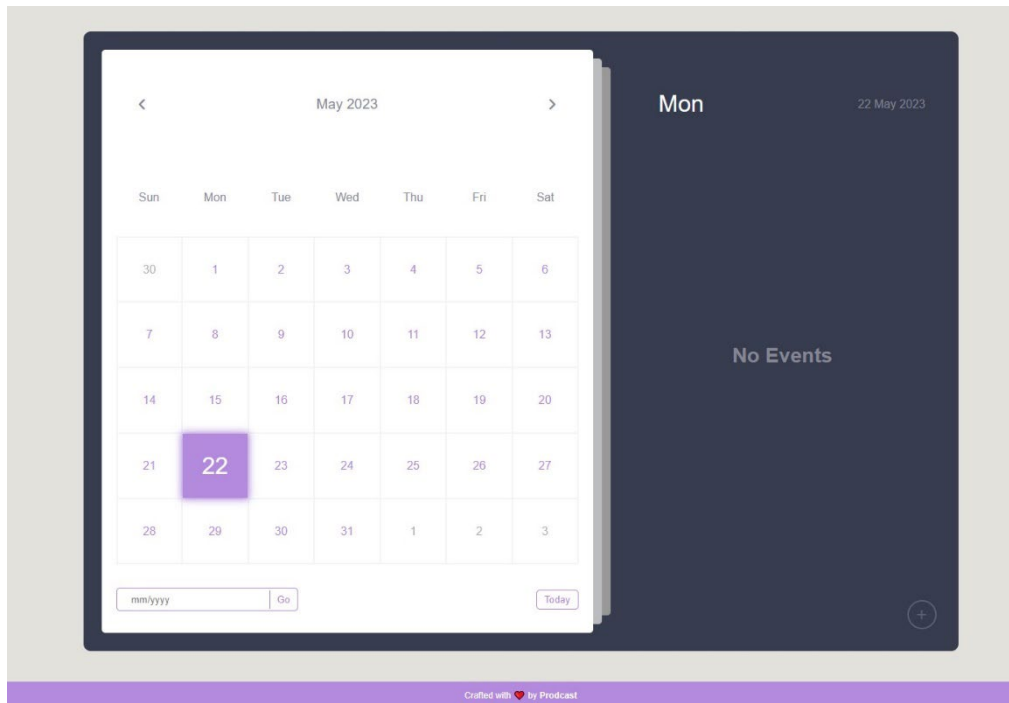


Figure 5

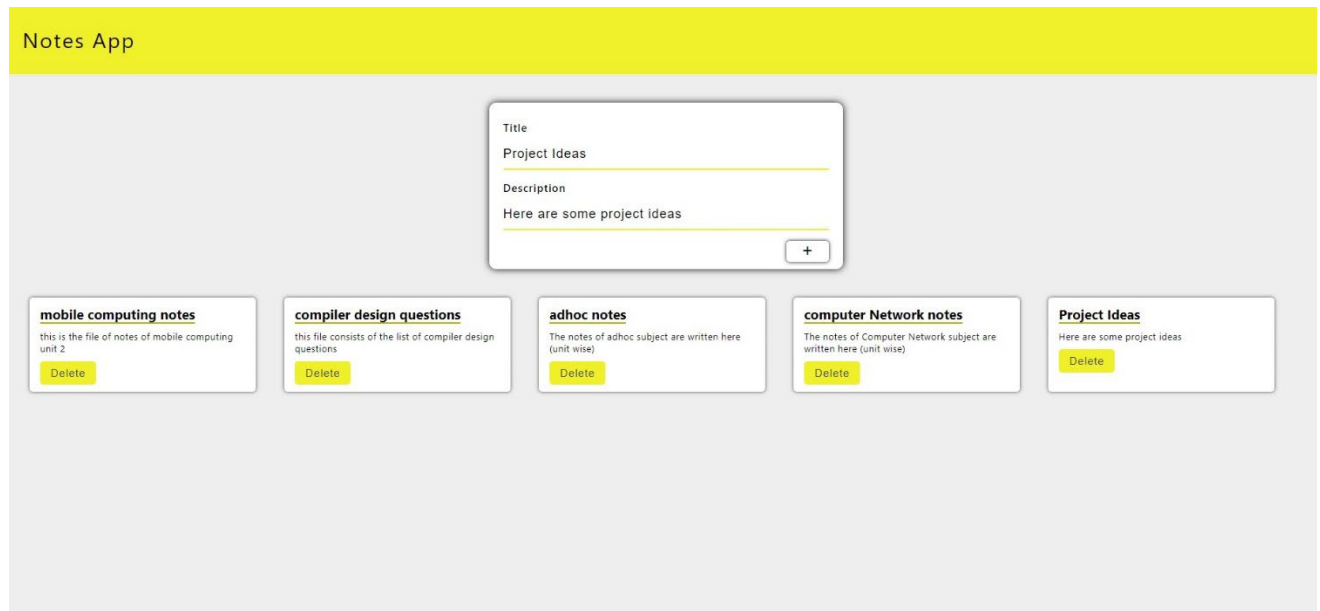


Figure 6

Chapter 6

Future Scopes and Limitations

Limitations

No software is perfect. All software has some limitations or cons that may be due to the problem domain it targets or the design of the system. This is no exception.

- Communication between various components and data binding can be painful
- Code reusability of views and view models is difficult.
- Managing view models and their state in nested views and complex UI's is difficult.
- Cost of operating an app costs money. Moreover the cost of running a public facing server that serves the script files will be considerably higher as this server will need to handle large amounts of concurrent requests. Therefore we cannot provide this platform as a free service

Future Scope

While the platform in its current form provides basic web analytics, many more features are expected to be present in a modern web analytics platform.

Some of these features include:

- Integrations with government resources for creating awareness among colleges.
- More ways of signing in to accounts like using college own authentication methods.
- Detailed communication between government agencies and colleges can be implemented.
- Some recommendations using machine learning can be generated on such vast data.

Chapter 7

Test Cases

Test Cases

Test case is an object for execution for other modules in the architecture that does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:-manual and automated. A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any, is expected to trigger errors.

Test Case ID	Test Case Name	Test Case Description	Step	Expected Result	Actual Result	Test Case Status
1	Signup	Test the signup button	Click on the signup button	A user should be able to sign up	Signup Successful	Pass
2	Login	Test the login functionality	Click on signup button	A user should be able to login.	Login Successful	Pass

3	Logout	Test the logout functionality	Click on logout button	A user should be able to redirect to the login screen.	Logout Successful	Pass
4	Post query	People should be able to post their tasks.	Click on add task.	Data should be added to the database.	Data added successfully	Pass
5	Get query	People must be able to retrieve their data.	Click on the login button.	Data should be presented on the dashboard.	Got data successfully	Pass
6	Post member	People must be able to add members to their project	Click on the add member button.	People can add members to their project.	Members added successfully.	Pass

Table 1

Chapter 8

Conclusion

Conclusion

To conclude, this project is a productivity tool used to track daily life data with a database without any limitations.

This can be used by any person but it would be beneficial for the students as working people who can improve their productivity. The application has a very smooth working process and is simple to operate the platform. Though the application needs to be developed further to make it function flawlessly, it still captures the essence of making an easy-to-use platform.

Chapter 9

Bibliography

Bibliography

The following resources were used as reference material for this project.

Product Documentation

1. HTML

- Mozilla Developer Network - <https://developer.mozilla.org>
- <https://www.youtube.com/watch?v=GpcMasRWUhl&list=PL0b6OzIxLPbxStBQ21C2toa5uQMqHEoRT>

2. CSS

- Mozilla Developer Network - <https://developer.mozilla.org>
- <https://www.youtube.com/watch?v=TThZIt4r3eg&list=PL0b6OzIxLPbzDsI5YXUa01QzxOWyqmrWw>

3. JS

- Mozilla Developer Network - <https://developer.mozilla.org>
- https://www.youtube.com/watch?v=Lgxgm-T9cgA&list=PL0b6OzIxLPbx-BZTaWu_AF7hsKo_Fvsnf

4. REACT.JS

- <https://www.youtube.com/watch?v=LX4JUscM9Sk&list=PLjVLYmrlmjGdnIQKgnTeR1T9-1ltJEaJh>

5. TAILWIND

- https://www.youtube.com/watch?v=6LhRLZ5_IgI&list=PLjVLYmrlmjGfpwYhVAbiGAhFl6h8XWDV

Chapter 10

Appendix

In this appendix, you will find a collection of additional resources and information to supplement your experience with our productivity app. These resources include user guides. The appendix serves as a handy reference to assist you in fully utilizing the app's features and overcoming any challenges you may encounter. Whether you need step-by-step instructions or troubleshooting tips, the resources in this appendix are designed to enhance your productivity app experience and ensure you get the most out of it.

User guide:

- **Creating an account:** To use the productivity website, you may need to create an account by providing your email address and choosing a password.
- **Navigation:** Once you have logged in, you can navigate through the different sections of the website, such as the todo list, stopwatch, news section, notes section, clock, and calendar section.
- **Adding tasks:** To add tasks to your todo list, click on the "Add Task" button and enter the task details, such as the task name, due date, priority level, and any additional notes.
- **Reading news:** To access the news section, click on the "News" tab and browse through the latest news articles related to productivity and time management.
- **Taking notes:** To take notes, click on the "Notes" tab and start typing your notes. You can also format your notes by using bold, italic, or underline text.