FULL STACK PROJECT REPORT

**On**

**“KEEPER”**

**Submitted by-**

**Aditya Pratap Singh**

**Roll No: 191500060**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**2022-2023**

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**



**Declaration**

I hereby declare that the work which is being presented in the Full Stack Project “**KEEPER”,** in partial fulfillment of the requirements for Full Stack Project viva voce, is an authentic record of my own work carried by me under the supervision of our mentor Mr. Mandeep Singh.

Member: Aditya Pratap Singh (191500060)

Course: Bachelor of Technology (Computer Science and Engineering)

Year: 3rd

Semester: 6th

## Supervised By:

Mr. Mandeep Singh, Assistant Professor,

GLA University, Department of Computer Engineering & Application

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**



**Certificate**

This is to certify that the above statements made by the candidate is correct to the best of my/our knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor

Mr. Mandeep Singh

Technical Trainer

Dept of CEA, GLA University

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Coordinator Program Coordinator

(Mr. Mayank Srivastava) (Mr. Shashi Shekar)

**About the Project**

This full-stack project “Keeper” is a note-taking service included as part of the free, web-based Docs Editors suite. Keeper allows users to record plain-text notes and organize, edit, or share them with others using a suite of collaboration tools. You can also use Keeper app to create, set time- and note-based reminders. You will in the end getting your productivity increased of doing tasks, that will help you to utilize your time in the best way possible. You just have to visit the site and add the noes and tasks according to the need and all those tasks will be added to the dashboard over there. The website is very simple to use and has a very simple yet good UI. Additionally, it will help everyone to get into a proper day to day routine of managing your tasks properly. Not only this, but Keeper’s objective also is to provide a platform to all those who are bad at time management and tend to forgot petty important things.

**Motivation**

Keeper’s objective is to provide such an interface that can help user to store all notes and essential information digitally, usually in a cloud-based storage system. Type, write, and draw notes on the device of choice just as one would using pen and paper. Successful people have notes and tasks maintained at one place as well. That’s because they have a clear purpose that aligns the daily grind with their long-term aspirations.

Too many of us get caught up in the madness of today and forget to look out into our future. I know because I speak with lots of smart candidates about joining our company and most really have not thought about where they are going.

Developing a personal roadmap has four major benefits:

* It drives self-awareness.
* It provides a framework for success and a true north to keep working towards.
* It provides insight into which skills and resources you need to reach your goals.
* It makes it easier to ask for assistance from others on your journey.

Not only do reminder apps keep you on track throughout a busy day – but they also do great things for your brain and productivity levels. You can focus on giving one task at a time your full attention, safe in the knowledge that you'll know exactly what you need to do next.

**Requirements**

**a). Software Requirements:**

* Technology Implemented: Full-Stack Technologies
* Languages/Technologies Used: HTML, CSS, JavaScript, Motoko
* IDE Used: Visual Studio Code
* JavaScript Library Used: React
* Web Browser: Google Chrome
* GitHub: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub Repository: A GitHub repository can be used to store a development project. It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images). A GitHub repository should also include a license file and a README file about the project. A GitHub repository can also be used to store ideas, or any resources that you want to share.
* Visual Studio Code: Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Microsoft has released Visual Studio Code's source code on the VS Code repository of GitHub.com, under the permissive MIT License, while the compiled binaries are freeware.

**b). Hardware Requirements:**

* Processor Required: Intel i5/ Ryzen 5
* Operating System: Windows 10
* RAM: 4GB (minimum)
* Hardware Devices: Computer System
* Hard Disk: 256GB

**Acknowledgement**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to my mentor Mr. Mandeep Singh for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

I would like to express my gratitude to my helping peers for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to organization persons for giving me such attention and time.

My thanks and appreciations also go to my helping peers in developing the project and people who have willingly helped me out with their abilities.

**KEEPER**

**Abstract**

“Keeper” is a note-taking service included as part of the free, web-based Docs Editors suite. Keeper allows users to record plain-text notes and organize, edit, or share them with others using a suite of collaboration tools. You can also use Keeper app to create, set time- and note-based reminders. You will in the end getting your productivity increased of doing tasks, that will help you to utilize your time in the best way possible. You just must visit the site and add the noes and tasks according to the need and all those tasks will be added to the dashboard over there. The website is very simple to use and has a very simple yet good UI. Additionally, it will help everyone to get into a proper day to day routine of managing your tasks properly. Not only this, but Keeper’s objective also is to provide a platform to all those who are bad at time management and tend to forget petty important things.

Not only do reminder apps keep you on track throughout a busy day – but they also do remarkable things for your brain and productivity levels. You can focus on giving one task at a time your full attention, safe in the knowledge that you will know exactly what you need to do next.

**Contents**

**Acknowledgment…………………………………….........**08

**Abstract…………………………………………………**09

**Introduction:**

Introduction to Full-Stack Technologies........…......12-13

Pre-requisites………………………….14

1. **Technologies Used:**

Front-End Technologies……………………………….15

Back-End Technologies……………………......................16

ReactJS………………………………..............17-18

NodeJS...............................................................19

**3. List of Figures...……………………………………**20-26

**4. Software Testing..............................................................**27-31

**5.Conclusion……………………………………………**32

**6.Bibliography………………………………………….**33

**Chapter 1**

**Introduction**

Most of us tend to forget the important things or tasks because of our daily hectic schedule. To overcome, this problem, Keeper is an exceptionally good solution. This web app allows you to store all the important things at one place in written form, and will also keep you motivated throughout the day, just to complete those unfinished tasks. In the end, your productivity level will get start augmenting. Successful people have notes and tasks maintained at one place as well. That’s because they have a clear purpose that aligns the daily grind with their long-term aspirations.

Too many of us get caught up in the madness of today and forget to look out into our future. I know because I speak with lots of smart candidates about joining our company and most really have not thought about where they are going.

“Productivity is being able to do things that you were never able to do before.”

**Chapter 2**

**Technologies Used**

**HTML**

HTML is the standard markup language for creating Web pages. Every website you open in your web browser, from social networks to music services, uses HTML. A look under the hood of any website would reveal a basic HTML code page, written with an HTML structure editor, providing structure for all the page’s components, including its header element, footer element, main content, and other inline elements. When using HTML for your website, you get to control everything on your website. Because you build your website from the core using a markup language, accessing, and modifying the code is easier than with WordPress. This makes it much more flexible when integrating certain additional features.

**CSS**

CSS is the language we use to style a Web page. With CSS, we can create rules, and apply those rules to many elements within the website. This approach offers many advantages when a client requires site-wide changes. Since the content is completely separated from the design, we can make those changes in our Style Sheet and have it effect every applicable instance.

Since rules are only downloaded once by the browser, then are cached and used for each page load, the use of CSS can lead to lighter page loads, and improved performance.

This contributes to lighter server load and lower requirements, which overall saves money for our clients. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

**JavaScript**

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

JavaScript enables dynamic interactivity on websites when it is applied to an HTML document. JavaScript helps the users to build modern web applications to interact directly without reloading the page every time. JavaScript is commonly used to dynamically modify HTML and CSS to update a user interface by the DOM API. It is mainly used in web applications.

I have also used the JavaScript library React. React JS is basically a JavaScript library built and maintained by Facebook. React is an efficient, declarative, and flexible open-source JavaScript library for building simple, fast, and scalable frontends of web applications.

**Motoko**

The Motoko programming language is a new, modern and type safe language for developers who want to build the next generation of distributed applications to run on the Internet Computer blockchain network.

Motoko provides a rich set of abstractions, data types, and concurrency primitives that allow developers to create sophisticated applications without having to deal with lower-level infrastructure concerns such as message passing, memory management, or garbage collection.

I have used this new programming language to persist the data of my project on my local computer. It works so seamlessly.

**Pre-requisite**

Hands-on knowledge of JavaScript, HTML and CSS is essential before working on the concepts for making of webpages. Make sure that you have the browser or chrome installed and running before opening website.

Knowledge of some back-end part is must. JavaScript and its libraries React is used in this project.

And I’ve used this new programming language to persist the data of my project on my local computer.

And knowledge about how to use GitHub is a must, and Git commands. Also, we must maintain our whole project on GitHub, so knowledge of it is very necessary.

**Chapter 3**

**List of Figures**

* **THE Main Page**

Graphical user interface, text, application

Description automatically generated

* **Roadmap Section**
* **Title & Note Content Section**

Graphical user interface

Description automatically generated with medium confidence

* **This is how Title & Content looks like before pressing ADD (+) button**

Graphical user interface, text

Description automatically generated

* **After hitting (+), the note with its title has been added to the queue.**

Graphical user interface

Description automatically generated with medium confidence

* **You can add as my tasks or notes you want :)**

Graphical user interface, website

Description automatically generated

* **After refreshing also, the data will persist.**

Graphical user interface

Description automatically generated



* **After finishing, to Delete the task or your notes, just**

**simply hit this (Delete) button.**

Graphical user interface, text

Description automatically generated



* **After Deleting, if you refresh the website, the deleted data will not be shown again.**

Graphical user interface, text

Description automatically generated

**Chapter 4**

**Software Testing**

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing includes designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques. System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data. In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing, and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity, and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking. Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages. The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation.

4.1 TERMINOLOGY

Error The term error is used in two ways. It refers to the difference between the actual output of software and the correct output, in this interpretation, error is essential a measure of the difference between actual and ideal. Error is also to use to refer to human action that result in software containing a defect or fault.

Fault is a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug.

Failure is the inability of a system or component to perform a required function according to its specifications. A software failure occurs if the behavior of the software is the different from the specified behavior. Failure may be caused due to functional or performance reasons.

4.2 TYPES OF TESTING

**a. Unit Testing** The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system. A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested separately, independent of each other. Since the check is done at the program level, it is also called program teasing.

**b. Module Testing** A module and encapsulates related component. So can be tested without other system module.

**c. Subsystem Testing** Subsystem testing may be independently design and implemented common problems are sub-system interface mistake in this checking we concenton it. There are four categories of tests that a programmer will typically perform on a program unit.

i Functional test

ii Performance test

iii Stress test

iv Structure test

**Functional Test** Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

**Performance Test** Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit. A certain amount of avoid expending too much effort on fine-tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels.

**Stress Test** Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

**Structure Test** Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing. While structure testing is referred to as “white box” or “glass box” testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring the test coverage achieved when the test cases are exercised.

**Chapter 5**

**Conclusion**

I have completed our project within time limit with the help of some of my helping peers under the supervision of our mentor Mr. Mandeep Singh.

My GitHub project link is-

<https://github.com/AdityaPSingh3/Keeper>

**Chapter 6**

**Bibliography**

**developer.mozilla.org**

[**www.google.com**](http://www.google.com)

**www.geeksforgeeks.org**

[**www.youtube.com**](http://www.youtube.com)

**www.w3schools.com**

[**www.smartcontracts.com**](https://smartcontracts.org/)

**www.devdocs.io**