**PROJECT REPORT**

# on

**PluginVerse**

# for

**Digipodium**

# Towards partial fulfillment of the requirement for the award of degree of

**Bachelor of Computer Applications**

# from

Babu Banarasi Das University Lucknow



**Academic Session 2020 – 21**

**School of Computer Applications**

|  |
| --- |
| **I Floor, H-Block, BBDU, BBD City, Faizabad Road, Lucknow (U. P.) INDIA 226028**  **PHONE: HEAD:** 0522-3911127, 3911321 **Dept. Adm. & Exam Cell:** 0522-3911326 **Dept. T&P Cell:** 0522-3911128; **E-Mail:** [head.sca@gmail.com](mailto:head.sca@gmail.com) |
| **w w w . b b d u . a c . i n** |

**PROJECT REPORT**

# on

**PluginVerse**

# for

**Digipodium**

# towards partial fulfillment of the requirement for the award of degree of

**Bachelor of Computer Applications**

# from

**Babu Banarasi Das University Lucknow**

# Developed and Submitted by Under Guidance of

Samarth Rai Mohammad Mubassir 1200211

Deepak Chandra Maurya

1200211067

**Academic Session 2020 - 21**

**School of Computer Applications**

|  |
| --- |
| **I Floor, H-Block, BBDU, BBD City, Faizabad Road, Lucknow (U. P.) INDIA 226028**  **PHONE: HEAD:** 0522-3911127, 3911321 **Dept. Adm. & Exam Cell:** 0522-3911326 **Dept. T&P Cell:** 0522-3911128; **E-Mail:** [head.sca@gmail.com](mailto:head.sca@gmail.com) |
|  |
| [**w w w . b b d u . a c . i n**](http://www.bbdu.ac.in/) |

|  |  |
| --- | --- |
| **Babu Banarasi Das University Lucknow**  **CERTIFICATE**  **This is to certify that Project Report entitled**  **PluginVerse**  **being submitted by**  **Samarth Rai , Deepak Chandra Maurya**  **towards the partial fulfillment of the requirement for the award of the degree of**  **Bachelor of Computer Applications**  **to**  **Babu Banarasi Das University Lucknow**  **in the Academic Year 2020-21**  **is a record of the student’s own work carried out at**  **Digipodium**  **and to the best of our knowledge the work reported herein does not form a part of any other thesis or work on the basis of which degree or award was conferred on an earlier occasion to this or any other candidate.**  **Prabhash Ch. Pathak**  **HEAD (School of Computer Applications)** |  |
| 3 |

|  |  |
| --- | --- |
| **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 DIGIPODIUM INSTITUTE for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.  I would like to express my gratitude towards my friends & member of DIGIPODIUM 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 group members for giving me such attention and time.  My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities. |  |
| 4 |

|  |  |
| --- | --- |
| **DECLARATION**  I **Samarth Rai & Deepak Chandra Maurya** hereby declare that this project report entitled **PluginVerse ,** submitted by us, under the guidance of Mohammad Mubassir of **DIGIPODIUM INSTITUTE, LUCKNOW** is our own and has not been submitted to any other University or Institute or published earlier.  **Signature of Student : Samarth Rai**  **Deepak Chandra Maurya**  **BCA VIth Semester Date: 9/06/2023** |  |
| 5 |

|  |  |
| --- | --- |
| **ABSTRACT**   * Plugins are code components that can be added to an existing program or application to extend its functionality. * Plugin generator(PluginVerse) are commonly used to add specific features of capabilities to a website. * A website plugin could be used to add a contact form, social media sharing buttons, or an ecommerce shopping cart. |  |
| 6 |

|  |  |
| --- | --- |
| **TABLE OF CONTENTS**   1. Introduction of the Project    1. Introduction 2. Need of Identification    1. Objective    2. Scope 3. Problem Statement 4. System Analysis    1. Iterative waterfall Model    2. Feasibility Study    3. Database Design    4. Class Diagrams    5. Use Case Diagrams    6. Modules    7. Module Description    8. Gantt Chart 5. Software Requirement Specification(SRS)    1. Software Requirement    2. Hardware Requirement    3. Functional Requirement    4. Non Functional Requirement |  |
| 7 |

|  |  |
| --- | --- |
| 1. Tools & Technology 2. Cost estimation and cost constructive model 3. Testing    1. Testing Phase    2. Level of testing    3. Verification And Validation(V&V) 4. Screen Shots 5. Future Scope 6. Conclusion |  |
| 8 |

|  |  |
| --- | --- |
| **INTRODUCTION OF THE PROJECT**  Plugins are code components that can be added to an existing program or application to extend its functionality. In the context of websites, plugins are commonly used to add specific features or capabilities to a website that are not part of the website's core functionality.  For example, a website plugin could be used to add a contact form, social media sharing buttons, or an e-commerce shopping cart.  Plugins are used to add specific features and functionalities to websites that are not included in their core functionality. Here are some of the main reasons why plugins are used:   1. Enhanced website functionality: Plugins can add a wide range of features and functionalities to a website, such as contact forms, social media sharing buttons, image galleries, and e-commerce functionality. 2. Time-saving: Using plugins can save time and development costs by allowing website owners and developers to quickly and easily add new features to their websites without having to build them from scratch. 3. Customization: Plugins can be customized and configured to fit the specific needs and preferences of a website, allowing for greater flexibility and control. 4. Accessibility: Plugins can be used by website owners and developers of all skill levels, including those with little to no programming experience. 5. Scalability: Plugins can be added or removed as needed, allowing a website to grow and evolve over time.   We provide a range of powerful functionalities to enhance your website, including an Authentication System to secure user access, a Cart System for seamless e-commerce transactions, a Store Management System for efficient inventory management, a Payment System to enable secure online transactions, a Subscription System for recurring payments, a Plugin Generator to customize your website's features, a Feedback Handler to manage customer feedback, a File Hosting System to store and manage files, an Order Management System to streamline your order processing, and a Dashboard Generator to easily track your website's performance.  Our Solution is more useful than others which are already available like: Woocommerce , shopify but these websites can only be used for making websites and their plugins could use to make website on wordpress not on other technologies like HTML CSS etc. Now , we provide a plugin generator website which can work across other technologies like MERN React , HTML CSS and many more |  |
| 9 |

|  |  |
| --- | --- |
| **Need of identification**  **Objective & Scope :**  **Enhanced website functionality :**  Plugins can add a wide range of features and functionalities to a website.  **Time-Saving :**  Using plugins can save time and development costs by allowing website owners .  **Customization -**  Plugins can be customized and configured to fit the specific needs allowing for greater flexibility  & control.  **Accessibility -**  Plugins can be used by website owners and developers.  **Scalability –**  Plugins can be added or removed as needed. |  |
| 10 |

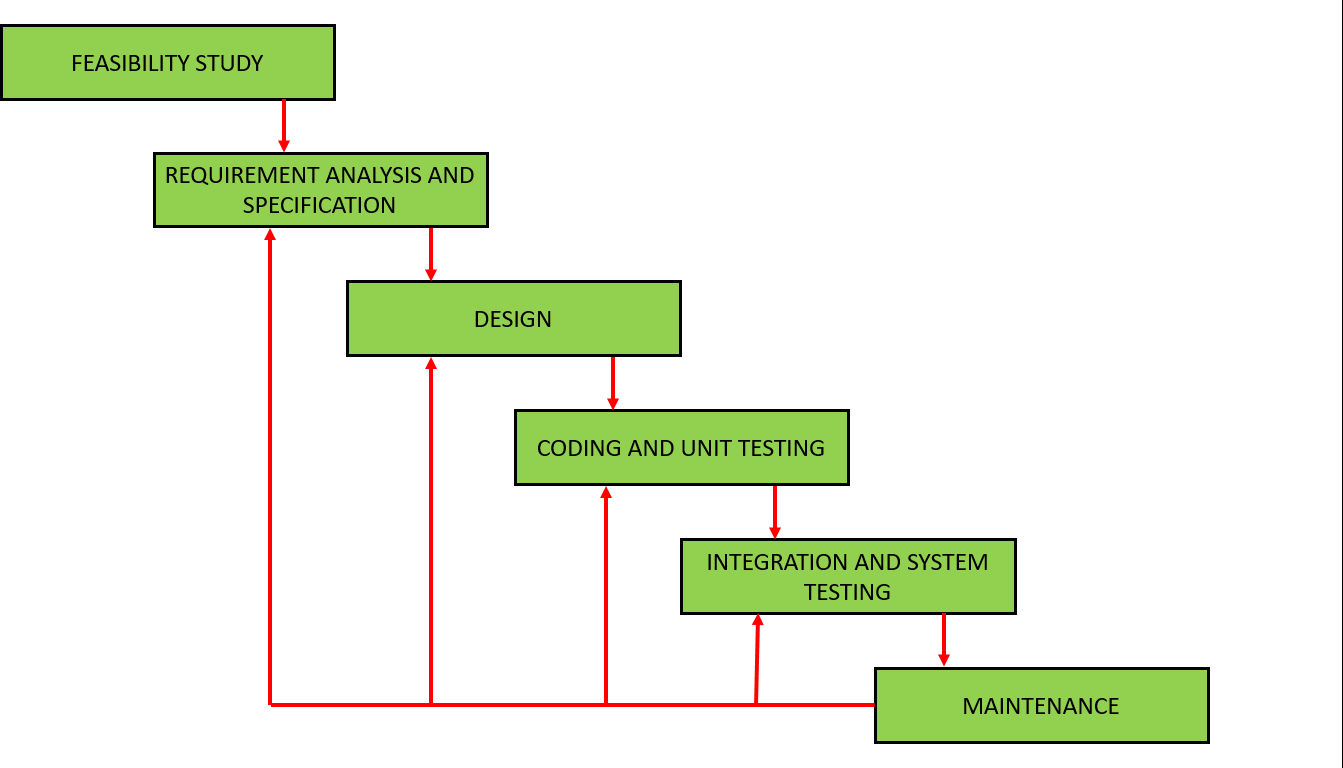
|  |  |
| --- | --- |
| **Proposed System**  **CREATE ACCOUNT -**  Create account/Login register.  **SELECT PLUGIN -**  Select plugin from Provided plugin.  **CONFIGURE PLUGIN -**  Plugins can be add remove and update .  **GENERATE PLUGIN CODE -**  Generate small line of code to use plugin.  **VIEW PLUGIN DOCUMENTATION -**  Give detail how to use specific plugin.  **ADD PLUGIN TO WEBSITE -**  Choose plugin and add to website . |  |
| 11 |

|  |  |
| --- | --- |
| **PROBLEM STATEMENT**   * Many website owners and developers struggle to find the right plugins. Finding plugins can be time-consuming, complicated, and expensive. * A plugin generator website is needed to enable users to create custom plugins quickly and easily and should not require extensive coding skills or resources |  |
| 12 |

|  |  |
| --- | --- |
| **System Analysis**  **Iterative waterfall model**  Iterative waterfall model assumes that one phase can start after completion of the previous phase, But in real projects, phases may overlap to reduce the effort and time needed to complete the project. Risk handling not supported – Projects may suffer from various types of risks.   * **Feedback Path –**   In the classical waterfall model, there are no feedback paths, so there is no mechanism for error correction. But in the iterative waterfall model feedback path from one phase to its preceding phase allows correcting the errors that are committed and these changes are reflected in the later phases.   * **Simple –**   Iterative waterfall model is very simple to understand and use. That’s why it is one  of the most widely used software development models.   * **Cost-Effective –**   It is highly cost-effective to change the plan or requirements in the model. Moreover, it is best suited for agile organizations.   * **Difficult to incorporate change requests –**   The major drawback of the iterative waterfall model is that all the requirements must be clearly stated before starting the development phase. Customers may change requirements after some time but the iterative waterfall model does not leave any scope to incorporate change requests that are made after the development phase starts. |  |
| 13 |

|  |  |
| --- | --- |
| * **Incremental delivery not supported –**   In the iterative waterfall model, the full software is completely developed and tested before delivery to the customer. There is no scope for any intermediate delivery. So, customers have to wait a long for getting the software.   * **Overlapping of phases not supported –**   Iterative waterfall model assumes that one phase can start after completion of the previous phase, But in real projects, phases may overlap to reduce the effort and time needed to complete the project.   * **Risk handling not supported –**   Projects may suffer from various types of risks. But, the Iterative waterfall model has no mechanism for risk handling.   * **Limited customer interactions –**   Customer interaction occurs at the start of the project at the time of requirement gathering and at project completion at the time of software delivery. These fewer interactions with the customers may lead to many problems as the finally developed software may differ from the customers’ actual requirements.   * **Well-organized -**   In this model, less time is consumed on documenting and the team can spend more time on development and designing |  |
| 14 |

|  |  |
| --- | --- |
| **Fig. ( Iterative waterfall model )** |  |
| 15 |



|  |  |
| --- | --- |
| **Feasibility Study**  All projects are feasible given unlimited resources and infinite time. Unfortunately the development of computer-based system in many cases is more likely to be plagued by scarcity of resources and delivery date. Hence, we have made use the concept of reusability that is what Object Oriented Programming (OOP) is all about.  The feasibility report of the project holds the advantages and flexibility of the project. This is divided into three sections:-   1. Operational Feasibility 2. Technical Feasibility 3. Economic Feasibility   **1.Operational Feasibility:-**  It determines how acceptable the software is within the organization. The evaluations must then determine the general attitude and skills. Such restriction of the job will be acceptable. To the users are enough to run the proposed budget, hence the system is supposed to the feasible regarding all except of feasibility. In operational Feasibility, we attempt to ensure that every user can access the system easily.  Operational feasibility of the project also exists because in today’s world most of the people are using the internet and are purchasing the products online. There is nothing complex in the system that cannot be used by people. It is socially accessible feasible as well because of its |  |
| 16 |

|  |  |
| --- | --- |
| usefulness and easiness in getting information. Time feasibility also exists because it can be developed and implemented in the given time. As far as legal feasibility is concerned there is no such restriction faced by the system.  **2. Technical Feasibility:-**  Technical feasibility centers on the existing computer system (Hardware and Software etc.) and to what extend it support the proposed addition. In this project, all the necessary cautions have been taken care to make it technically feasible. Using a key the display of text/object is very fast. Also, the tools, operating system and programming language used in this localization process is compatible with the existing one. The technical needs of the system vary considerably but might include:-   1. The facility to produce outputs in a given time. 2. Response time under certain conditions.   The project is technical feasible because of the availability of the required software hardware and technology. The changes can be made be made in the system as and when required.  **3. . Economic Feasibility:-**  Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to be determining the benefits and savings that are expected from a candidate and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. A systems financial benefit must exceed the cost of developing that system. i.e. a new system being developed should be a good investment for the organization. Economic feasibility considers the following:   1. The cost to conduct a full system investigation. 2. The cost of hardware and software for the class of application. 3. The benefits in the form of reduced cost or fewer costly errors. |  |
| 17 |

|  |  |
| --- | --- |
| **Data Base Design**  Database is critical for all businesses. A good database does not allow any form of anomalies and stores only relevant information in an ordered manner. If a database has anomalies, it is affecting the efficiency and data integrity. For example, delete anomaly arise upon the deletion of a row which also forces other useful data to be lost. As such, the tables need to be normalized. This fulfils the last objective of ensuring data are accurate and retrieved correctly. Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information. The organization of data in database aims to achieve three major objectives: -   * Data integration * Data integrity * Data independence   **All Table List:** |  |
| 18 |

|  |  |
| --- | --- |
| **User Table:**  **Equipment Table:**  **Order Table:** |  |
| 19 |

|  |  |
| --- | --- |
| Column Name | Data Type |
| Id | objectId |
| Fullname | String |
| Email | String |
| Password | String |
| Age | Number |
| Avatar | String |

|  |  |
| --- | --- |
| Column Name | Data Type |
| Id | objectId |
| Name | String |
| Description | String |
| Features | String |
| Price | Number |
| Avatar | String |

|  |  |
| --- | --- |
| Column Name | Data Type |
| Id | objectId |
| User | objectId |
| Equipment | ObjectId |
| Created | Date |

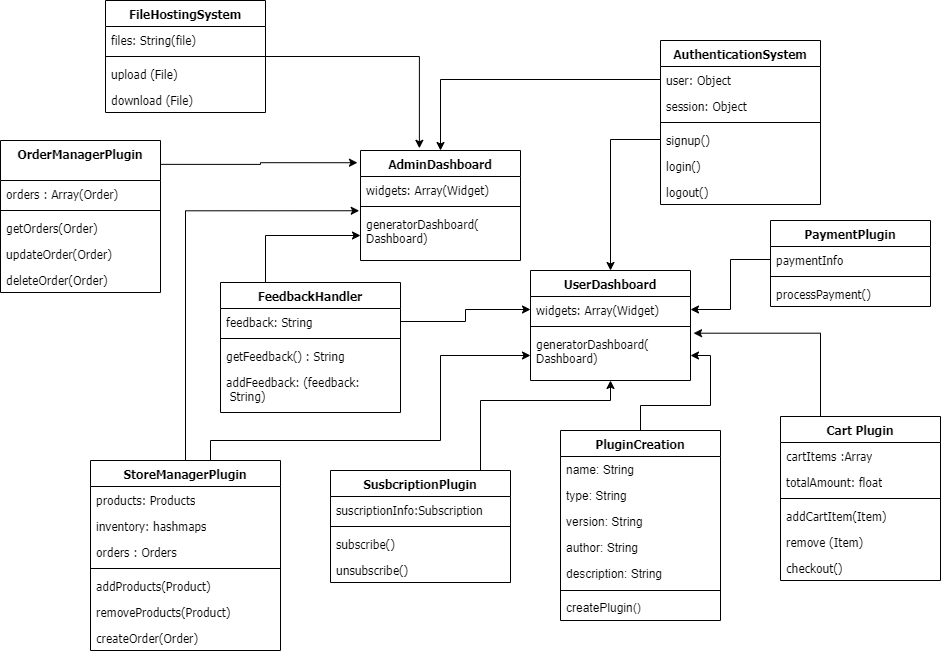
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Review Table:** | | | | |
|  | Column Name | Data Type |  | |
| Id | objectId |
| Rating | Number |
| User | ObjectId |
| Created | Date |
|  | **Use Case Diagram**  Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.  Hence to model the entire system, a number of use case diagrams are used. | |  | |
|  | | | | 20 |

|  |  |
| --- | --- |
| Purpose of Use Case Diagrams:-  The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.  Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.  When the initial task is complete, use case diagrams are modelled to present the outside view.  In brief, the purposes of use case diagrams can be said to be as follows −   * Used to gather the requirements of a system. * Used to get an outside view of a system. * Identify the external and internal factors influencing the system. * Show the interaction among the requirements are actors. |  |
| 21 |

|  |  |
| --- | --- |
| **Class Diagram**  Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.  Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.  Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.  Purpose of Class Diagrams:-  The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.  UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.  The purpose of the class diagram can be summarized as −   * Analysis and design of the static view of an application. * Describe responsibilities of a system. * Base for component and deployment diagrams. |  |
| 22 |

|  |  |
| --- | --- |
| * Forward and reverse engineering. |  |
|  |

**2**



# Modules

1. Authentication System
2. Cart System
3. Store Management System
4. Payment System
5. Subscription System.
6. Plugin Generator
7. Feedback Handler
8. File Hosting System
9. Order Management System
10. Dashboard Generator.

|  |  |
| --- | --- |
| In this project, three members are involved and so, we have divided the whole project into three groups -  **The following modules are handled by Samarth Rai -**   * **Module 1** - Plugin Generator * **Module 2** - Feedback Handler * **Module 3** - File Hosting System * **Module 4** - Order Management System * **Module 5** - Dashboard Generator. |  |
| 26 |

|  |  |
| --- | --- |
| **The following modules are handled by Deepak Chandra Maurya**   * **Module 6 –** Authentication System * **Module 7 –** Cart System * **Module 8 –**  Store Management System * **Module 9 –** Payment System * **Module 10 –** Subscription System |  |
| 27 |

|  |  |
| --- | --- |
| **Module Description**   1. **Module 1** - **Authentication System:**   Enables users to securely log in and access restricted areas of the website.   1. **Module 2** - **Plugin Generator:**   Enables developers to create custom plugins for a website.   1. **Module 3** - **Feedback Handler:**   Collects and manages user feedback and suggestions.   1. **Module 4** - **File Hosting System:**   Provides secure storage and sharing of files on a website.   1. **Module 5** - **Dashboard Generator:**   Generates customized dashboards for website analytics and performance tracking. |  |
| 28 |

|  |  |
| --- | --- |
| 1. **Module 6 – Cart Plugin:**   Allows customers to add products to a shopping cart and checkout when ready to purchase.   1. **Module 7 – Store Manager Plugin:**   Manages product listings, inventory, and orders for an e-commerce store.   1. **Module 8 – Payment Plugin:**   Processes online payments securely and efficiently.   1. **Module 9 – Subscription Plugin:**   Allows users to sign up for recurring payments and access exclusive content or services.   1. **Module 10 – Order Manager Plugin:**   Manages orders and shipping for an e-commerce store. |  |
| 29 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Gantt Chart**  A Gantt chart is popular type of chart that illustrates a project schedule. Gantt Chart illustrates the start and finish dates of the terminal elements and summary elements of a project. Terminal element and summary comprise the work breakdown structure of the project. | | | | | | | | | | | |
|  | **Task** | **4Feb-28Feb** | **28Feb-9Mar** | **10Mar-**  **12Apr** | | **13Apr-**  **16May** | | **17may-**  **22may** | **23may-**  **28may** | |  |
| **Develop project**  **proposal** | 27 days |  |  | |  | |  |  | |
| **Analysis** |  | 10 days |  | |  | |  |  | |
| **Designing** |  |  |  |  |  | |  |  | |
| 30 days | |
| **Coding** |  |  |  | |  |  |  |  | |
| 34days | |
| **Unit Testing** |  |  |  | |  | | 5 days |  | |
| **Implementation** |  |  |  | |  | |  |  |  |
| 5 days | |
|  | | | | | | | | | |  |
|  | | | | | | | | | | | 30 |

|  |  |
| --- | --- |
| **Software Requirement Specification(SRS) Software Requirement**  **Client Side**   * Web Browser (Google Chrome, Firefox, IE9 or above)   **Server Side**   * + Web Browser (Google Chrome, Firefox)   + Node JS 14 or above   + Vs code   + Frontend Framework – React 17   + Backend Framework – Express JS   + Database - MongoDB |  |
| 31 |

|  |  |
| --- | --- |
| **Hardware Requirements**  **CLIENT SIDE**  **DEVELOPER SIDE** |  |
|  |

|  |  |
| --- | --- |
| Processor | Dual Core or above |
| RAM | 1 GB or above |
| Disk space | 50 GB minimum |
| Monitor | 14” or above/tab/mobile |
| Others | Keyboard, mouse, Internet Connection |

|  |  |
| --- | --- |
| Processor | Quad Core or above |
| RAM | 4 GB or above |
| Disk space | 256 GB |
| Monitor | 14” or above |
| Others | Keyboard, mouse, Internet Connection |

# Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO** | **REQ. NO.** | **PRIORITY** | **REQUIREMENTS** |
| 1. | R1 | C | Register & Login |
| 2. | R2 | R | Search and view |
| 3. | R3 | O | Chat with experts |
| 4. | R4 | C | Provide products |
| 5. | R5 | C | Make payment |
| 6. | R6 | R | Implementation |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Non Functional Requirements** | | | | |
|  | **CATEGORY** | **NON FUNCTIONAL REQUIREMENTS** | |  |
| **Maintainability** | It easy is to be supported, changed, enhanced, and restructured over time | |
| **USABILITY** | Users must be open on the browser. | |
| **RELIABILITY** | App permission would be important and will not use any private data of users | |
| **PERFORMANCE** | The will able to give good perform on most of the OS | |
| **Security** | The system or its part will be protected against malware attacks or unauthorized access | |
| **Compatibility** | System integration is a must. As such, while a software program doesn’t necessarily need to be compatible with other systems to work, it does need compatibility to be useful to users. | |
|  |  | | |  |
|  | | | 34 | |

**Design & Implementation Constraints**

* Required any web browser i.e. chrome , opera , edge , etc.
* Internet connection should required.
* Required user authentication.
* User had to share it’s current location for home delivery of equipment
* User have to be authenticated in order make booking on this website

# Document convention

## Font Family

Garamond

* **Font Size** Heading - 16 Sub-heading -14 Paragraph - 12

## Font Color

Content-black

|  |  |
| --- | --- |
| **Tools & Technology**  **Web Browser -**  A web browser (commonly referred to as a browser or internet browser) is an [application](https://en.wikipedia.org/wiki/Application_software) [software](https://en.wikipedia.org/wiki/Application_software) for accessing the [World Wide Web.](https://en.wikipedia.org/wiki/World_Wide_Web) When a [user](https://en.wikipedia.org/wiki/User_(computing)) requests a [web page](https://en.wikipedia.org/wiki/Web_page) from a particular [website,](https://en.wikipedia.org/wiki/Website) the web browser retrieves the necessary content from a [web server](https://en.wikipedia.org/wiki/Web_server) and then displays the page on the user's device  **Html -**  The HyperText Markup Language, or HTML is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser.](https://en.wikipedia.org/wiki/Web_browser) It can be assisted by technologies such as [Cascading](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) [Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript.](https://en.wikipedia.org/wiki/JavaScript)  [Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a [web](https://en.wikipedia.org/wiki/Web_page) [page](https://en.wikipedia.org/wiki/Web_page) [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.  [HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links,](https://en.wikipedia.org/wiki/Hyperlink) quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets.](https://en.wikipedia.org/wiki/Bracket#Angle_brackets) Tags such as <img /> and <input /> directly introduce  content into the page. Other tags such as <p> surround and provide information about  document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.  HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript,](https://en.wikipedia.org/wiki/JavaScript) which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997 |  |
| 37 |

|  |  |
| --- | --- |
| **CSS -**  Cascading Style Sheets (CSS) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) such as [HTML](https://en.wikipedia.org/wiki/HTML)[.[1]](https://en.wikipedia.org/wiki/CSS#cite_note-1) CSS is a cornerstone technology of the [World Wide Web,](https://en.wikipedia.org/wiki/World_Wide_Web) alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript)[.[2]](https://en.wikipedia.org/wiki/CSS#cite_note-2)  CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors,](https://en.wikipedia.org/wiki/Color) and [fonts](https://en.wikipedia.org/wiki/Typeface)[.[3]](https://en.wikipedia.org/wiki/CSS#cite_note-3) This separation can improve content [accessibility,](https://en.wikipedia.org/wiki/Accessibility) provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be [cached](https://en.wikipedia.org/wiki/Cache_(computing)) to improve the page load speed between the pages that share the file and its formatting.  **JavaScript -**  JavaScript often abbreviated as JS, is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) specification. JavaScript is [high-level,](https://en.wikipedia.org/wiki/High-level_programming_language) often [just-in-time compiled,](https://en.wikipedia.org/wiki/Just-in-time_compilation) and [multi-](https://en.wikipedia.org/wiki/Programming_paradigm) [paradigm.](https://en.wikipedia.org/wiki/Programming_paradigm) It has [curly-bracket syntax,](https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#Curly-bracket_languages) [dynamic typing,](https://en.wikipedia.org/wiki/Dynamic_typing) [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) [object-orientation,](https://en.wikipedia.org/wiki/Object-oriented_programming) and [first-class functions.](https://en.wikipedia.org/wiki/First-class_function)  Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS,](https://en.wikipedia.org/wiki/CSS) JavaScript is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).  Over 97% of [websites](https://en.wikipedia.org/wiki/Website) use it [client-side](https://en.wikipedia.org/wiki/Client-side) for [web page](https://en.wikipedia.org/wiki/Web_page) behaviour,[[11]](https://en.wikipedia.org/wiki/JavaScript#cite_note-deployedstats-11) often incorporating third- party [libraries.](https://en.wikipedia.org/wiki/Library_(computing)) All major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute the code on the [user](https://en.wikipedia.org/wiki/User_(computing))'s device.  As a multi-paradigm language, JavaScript supports [event-driven,](https://en.wikipedia.org/wiki/Event-driven_programming) [functional,](https://en.wikipedia.org/wiki/Functional_programming) and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) [programming styles.](https://en.wikipedia.org/wiki/Programming_paradigm) It has [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs) for working with text, dates, [regular expressions,](https://en.wikipedia.org/wiki/Regular_expression) standard [data structures,](https://en.wikipedia.org/wiki/Data_structure) and the [Document Object](https://en.wikipedia.org/wiki/Document_Object_Model) [Model](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM).  The ECMAScript standard does not include any [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O), such as [networking,](https://en.wikipedia.org/wiki/Computer_network) [storage,](https://en.wikipedia.org/wiki/Data_storage) or [graphics](https://en.wikipedia.org/wiki/Computer_graphics) facilities. In practice, the web browser or other [runtime](https://en.wikipedia.org/wiki/Runtime_system) [system](https://en.wikipedia.org/wiki/Runtime_system) provides JavaScript APIs for I/O.  JavaScript engines were originally used only in web browsers, but they are now core components of [other](https://en.wikipedia.org/wiki/JavaScript#Other_usage) software systems, most notably [servers](https://en.wikipedia.org/wiki/Server_(computing)) and a variety of [applications.](https://en.wikipedia.org/wiki/Application_software) |  |
| 38 |

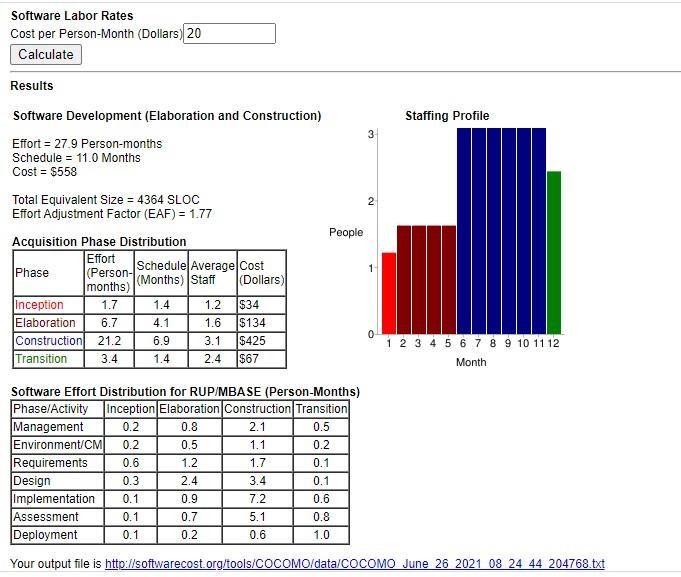
|  |  |
| --- | --- |
| **Node JS 14 -**  Node.js is an [open-source,](https://en.wikipedia.org/wiki/Open-source_software) [cross-platform,](https://en.wikipedia.org/wiki/Cross-platform) [back-end](https://en.wikipedia.org/wiki/Front_end_and_back_end) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [runtime environment](https://en.wikipedia.org/wiki/Runtime_system) that runs on the [V8 engine](https://en.wikipedia.org/wiki/V8_(JavaScript_engine)) and executes JavaScript code outside a [web browser.](https://en.wikipedia.org/wiki/Web_browser) Node.js lets developers use JavaScript to write command line tools and for [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting)—running scripts server- side to produce [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[[6]](https://en.wikipedia.org/wiki/Node.js#cite_note-6) unifying [web-](https://en.wikipedia.org/wiki/Web_application) [application](https://en.wikipedia.org/wiki/Web_application) development around a single programming language, rather than different languages for server-side and client-side scripts.  **Vs code -**  Visual Studio Code is a [source-code editor](https://en.wikipedia.org/wiki/Source-code_editor) made by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) for [Windows,](https://en.wikipedia.org/wiki/Windows) [Linux](https://en.wikipedia.org/wiki/Linux) and [macOS](https://en.wikipedia.org/wiki/MacOS)[.[9]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-TechCrunch-9) Features include support for [debugging,](https://en.wikipedia.org/wiki/Debugging) [syntax](https://en.wikipedia.org/wiki/Syntax_highlighting) [highlighting,](https://en.wikipedia.org/wiki/Syntax_highlighting) [intelligent code completion,](https://en.wikipedia.org/wiki/Intelligent_code_completion) [snippets,](https://en.wikipedia.org/wiki/Snippet_(programming)) [code refactoring,](https://en.wikipedia.org/wiki/Code_refactoring) and embedded [Git.](https://en.wikipedia.org/wiki/Git) Users can change the [theme,](https://en.wikipedia.org/wiki/Theme_(computing)) [keyboard shortcuts,](https://en.wikipedia.org/wiki/Keyboard_shortcut) preferences, and install [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)) that add additional functionality.  Microsoft has released most of Visual Studio Code's [source code](https://en.wikipedia.org/wiki/Source_code) on the microsoft/vscode repository of [GitHub](https://en.wikipedia.org/wiki/GitHub) using the "Code – OSS" name, under the permissive [MIT License](https://en.wikipedia.org/wiki/MIT_License)[,[5](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-code-oss-license-5)[][10]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-10) while the releases by Microsoft are proprietary [freeware](https://en.wikipedia.org/wiki/Freeware)  **React JS -**  React (also known as React.js or ReactJS) is an [open-source](https://en.wikipedia.org/wiki/Open-source) [front-end](https://en.wikipedia.org/wiki/Front_end_and_back_end) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library)[[3]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-react-3) for building [user interfaces](https://en.wikipedia.org/wiki/User_interfaces) or UI components. It is maintained by [Facebook](https://en.wikipedia.org/wiki/Facebook) and a community of individual developers and companies[.[4](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-4)[][5](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-5)[][6]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-6) React can be used as a base in the development of [single-page](https://en.wikipedia.org/wiki/Single-page_application) or mobile applications. However, React is only concerned with state management and rendering that state to the [DOM,](https://en.wikipedia.org/wiki/Document_Object_Model) so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality  **Express JS -**  Express.js, or simply Express, is a [back end](https://en.wikipedia.org/wiki/Front_end_and_back_end) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) for [Node.js,](https://en.wikipedia.org/wiki/Node.js) released as [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the [MIT License.](https://en.wikipedia.org/wiki/MIT_License) It is designed for building [web](https://en.wikipedia.org/wiki/Web_application) [applications](https://en.wikipedia.org/wiki/Web_application) and [APIs](https://en.wikipedia.org/wiki/API)[.[3]](https://en.wikipedia.org/wiki/Express.js#cite_note-ExpressJS-3) It has been called the [de facto standard](https://en.wikipedia.org/wiki/De_facto_standard) server framework for [Node.js](https://en.wikipedia.org/wiki/Node.js)[.[4]](https://en.wikipedia.org/wiki/Express.js#cite_note-4) |  |
| 39 |

|  |  |
| --- | --- |
| The original author, [TJ Holowaychuk,](https://en.wikipedia.org/w/index.php?title=TJ_Holowaychuk&action=edit&redlink=1) described it as a [Sinatra](https://en.wikipedia.org/wiki/Sinatra_(software))-inspired server,[[5]](https://en.wikipedia.org/wiki/Express.js#cite_note-5) meaning that it is relatively minimal with many features available as plugins. Express is the back-end component of popular development stacks like the [MEAN,](https://en.wikipedia.org/wiki/MEAN_(software_bundle)) [MERN](https://en.wikipedia.org/w/index.php?title=MERN&action=edit&redlink=1) or [MEVN](https://en.wikipedia.org/w/index.php?title=MEVN&action=edit&redlink=1) stack, together with the [MongoDB](https://en.wikipedia.org/wiki/MongoDB) database software and a [JavaScript](https://en.wikipedia.org/wiki/JavaScript) front-end framework or library  **MongoDB -**  MongoDB is a [source-available](https://en.wikipedia.org/wiki/Source-available) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [document-oriented database](https://en.wikipedia.org/wiki/Document-oriented_database) program. Classified as a [NoSQL](https://en.wikipedia.org/wiki/NoSQL) database program, MongoDB uses [JSON](https://en.wikipedia.org/wiki/JSON)-like documents with optional [schemas.](https://en.wikipedia.org/wiki/Database_schema) MongoDB is developed by [MongoDB Inc.](https://en.wikipedia.org/wiki/MongoDB_Inc) and licensed under the [Server Side Public](https://en.wikipedia.org/wiki/Server_Side_Public_License) [License](https://en.wikipedia.org/wiki/Server_Side_Public_License) (SSPL). |  |
| 40 |

|  |  |
| --- | --- |
| **Cost estimation and cost constructive model**  The proposed COCOMO (Constructive Cost Estimation Model) COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.  The key parameters which define the quality of any software products, which are also an outcome of the Cocomo, are primarily Effort & Schedule:   * **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units. * **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.   The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below  **Ei=a\*(KDLOC)b**  **Where** The value of the constant a and b are depends on the project type  **Basic COCOMO Model:** The basic COCOMO model provides an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:  **Effort=a1\*(KLOC)a2 PM Tdev=b1\*(efforts)b2 Months** Where  **KLOC** is the estimated size of the software product indicate in Kilo Lines of Code, a1,a2,b1,b2 are constants for each group of software products, |  |
| 41 |

|  |  |
| --- | --- |
| **Tdev** is the estimated time to develop the software, expressed in months,  **Effort** is the total effort required to develop the software product, expressed in **person months (PMs)**.  Thus in our project the following considerations is taken so as to get the cost estimation of this project. |  |
|  |

|  |  |
| --- | --- |
|  |  |
|  |

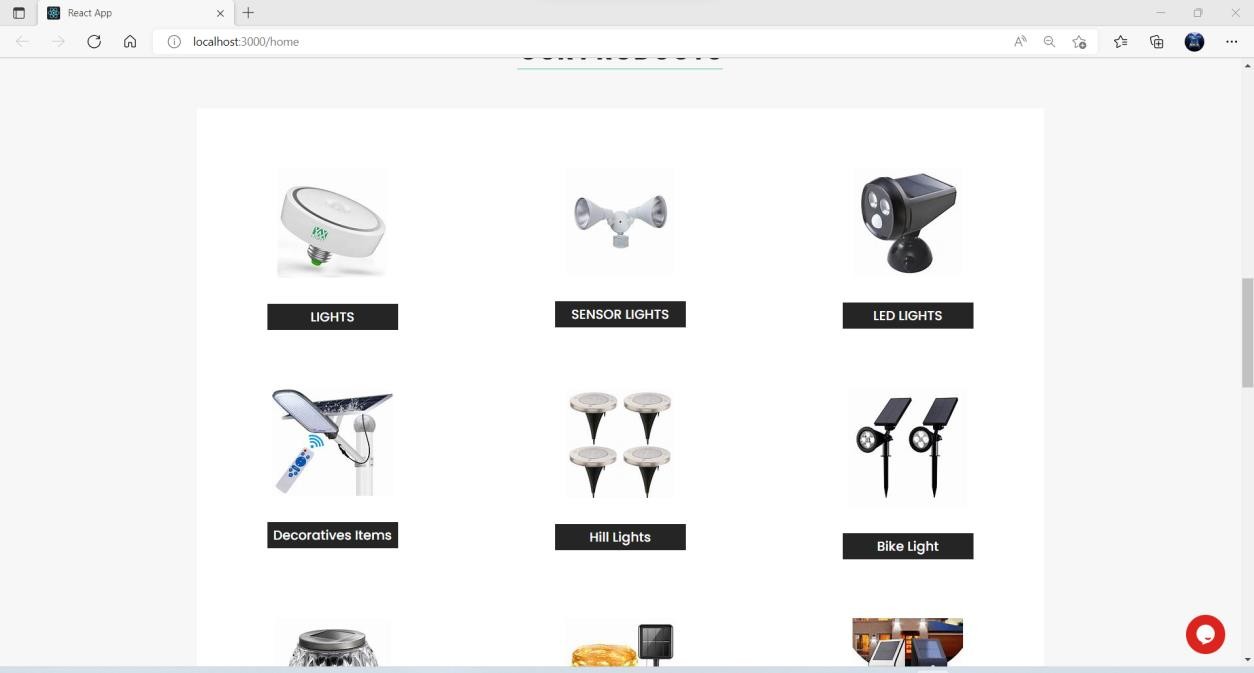


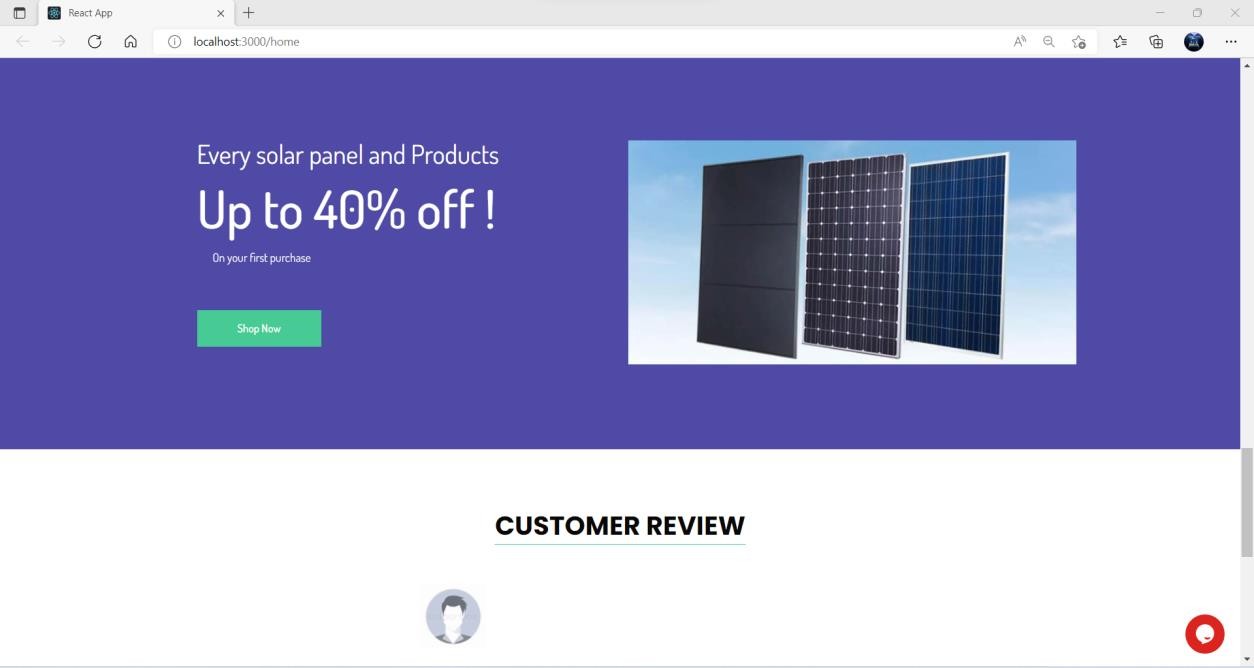
|  |  |
| --- | --- |
| **Testing**  **TESTING PHASE**  One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do. No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly design Lal system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.   1. Communication between the user and the designer. 2. The programmer's ability to generate a code that reflects exactly the system specification. 3. The time frame for the design.   Theoretically, a new designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the requirements of the user. The process of system testing and the steps taken to validate and prepare a system for final implementation are:  **LEVELS OF TESTING**  The different types of testing are as follow: 1.Unit Testing  2.Integration Testing 3.Acceptance testing  **1.Unit Testing:**  This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests. |  |
| 44 |

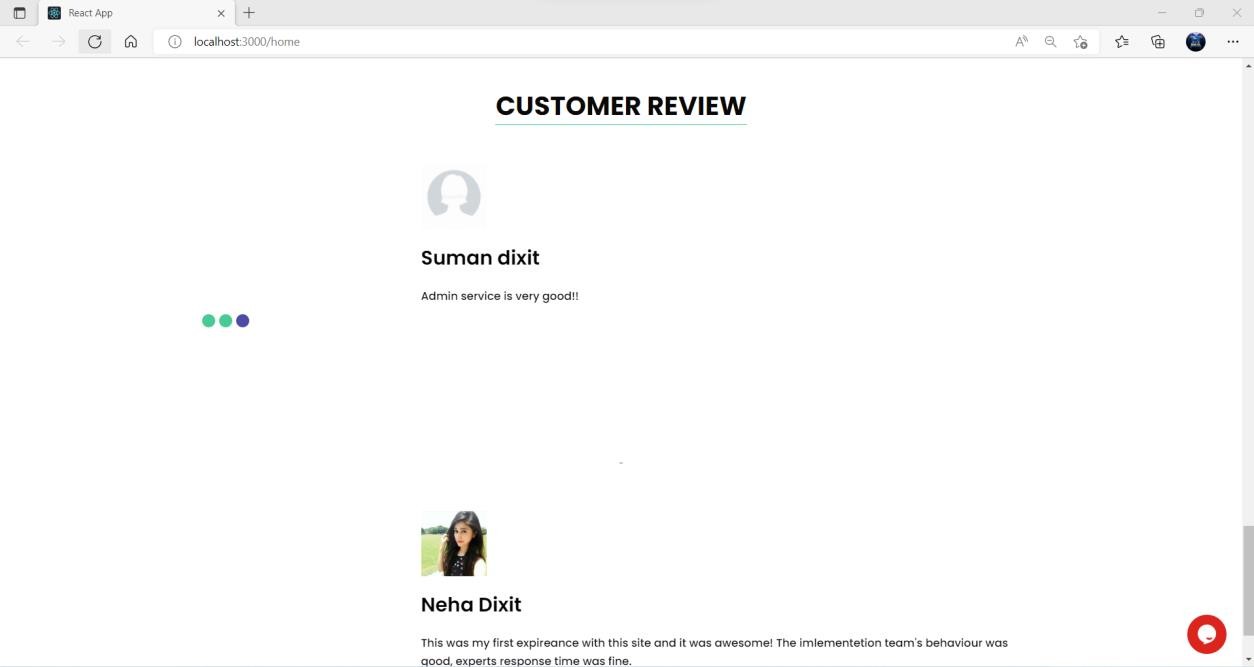
|  |  |
| --- | --- |
| 1. **Integration Testing:**   In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top- down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black hox testing technlques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.   1. **Acceptance testing:**   The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing- The screen layouts and output should already have been tested during the prototyping phase. An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested. The purpose of testing is to ensure that catch program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each lest run there is list of conditions tested, the test data used and the result expected. The test plan was then. reviewed to check that cache path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition |  |
| 45 |

|  |  |
| --- | --- |
| **VERIFICATION AND VALIDATION (V&V)**  The objectives of verification, validity nativities are to assess and improve the quality of the work products generated during development and modification of the Soll ware. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance standards. The terms verification and validation are used synonymously. These are defined an a pun Verification: "Are we building the product right?" Validation: "Are we building the right product?" Verification activities include proving. testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities. The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for cache V&V task, and criteria for evaluating the outputs are also specified.  The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g. trace ability and document evaluation). Due to this, inspections play a significant role in verification. |  |
| 46 |

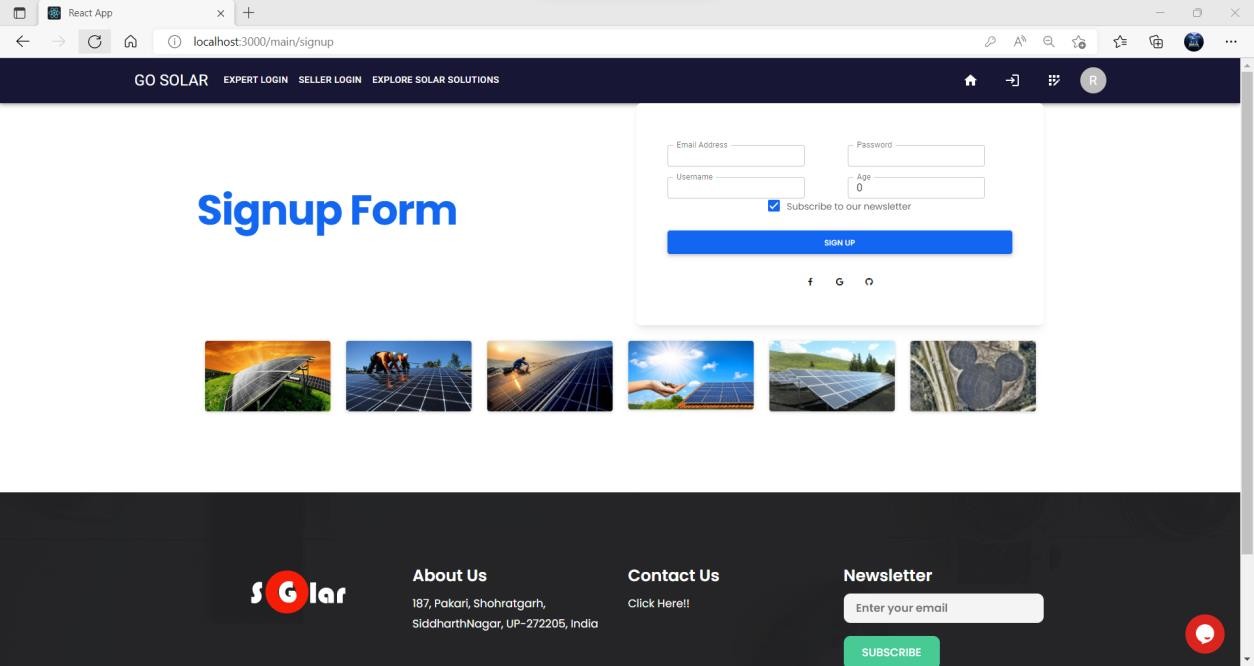
|  |  |
| --- | --- |
| **Screen Shots**  **Home Page -** |  |
|  |



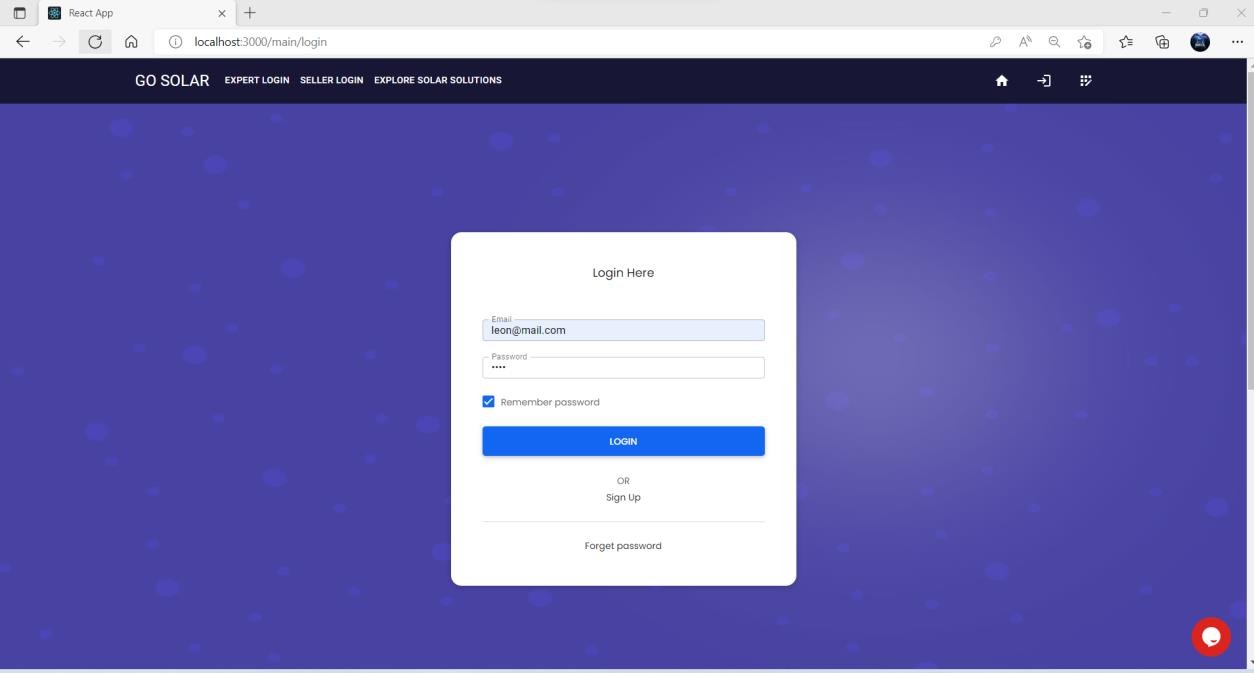




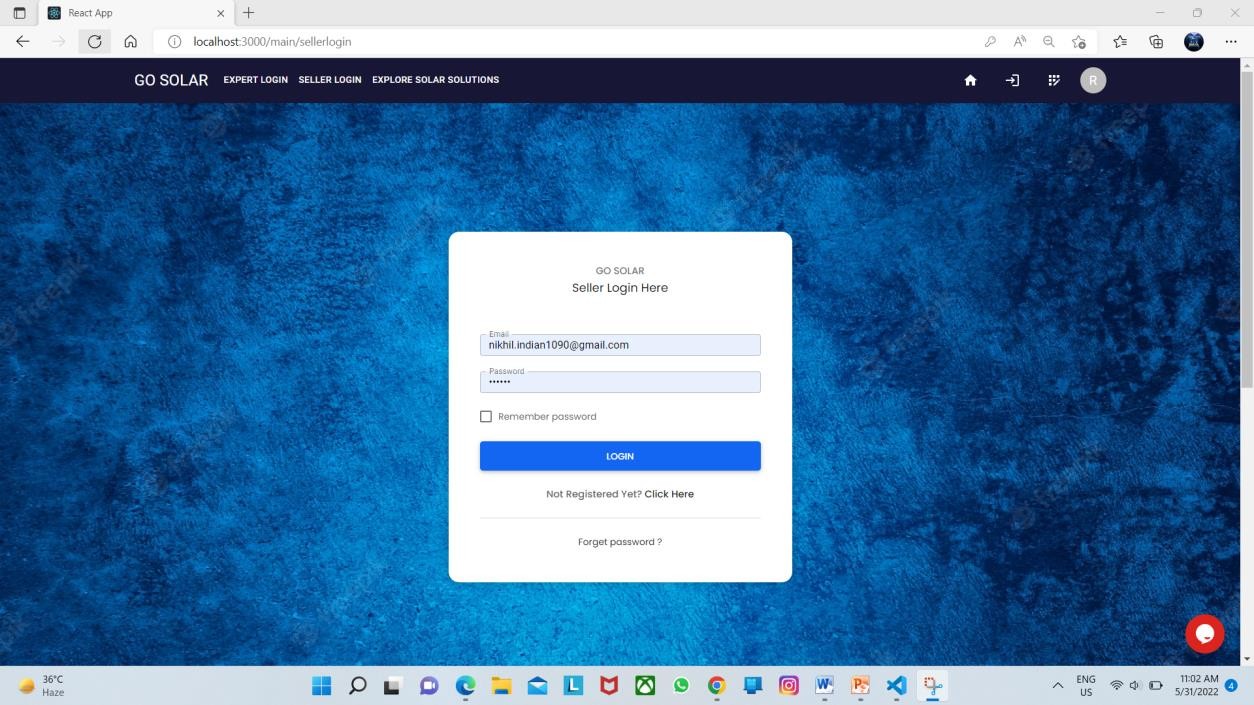
## Sign up page -

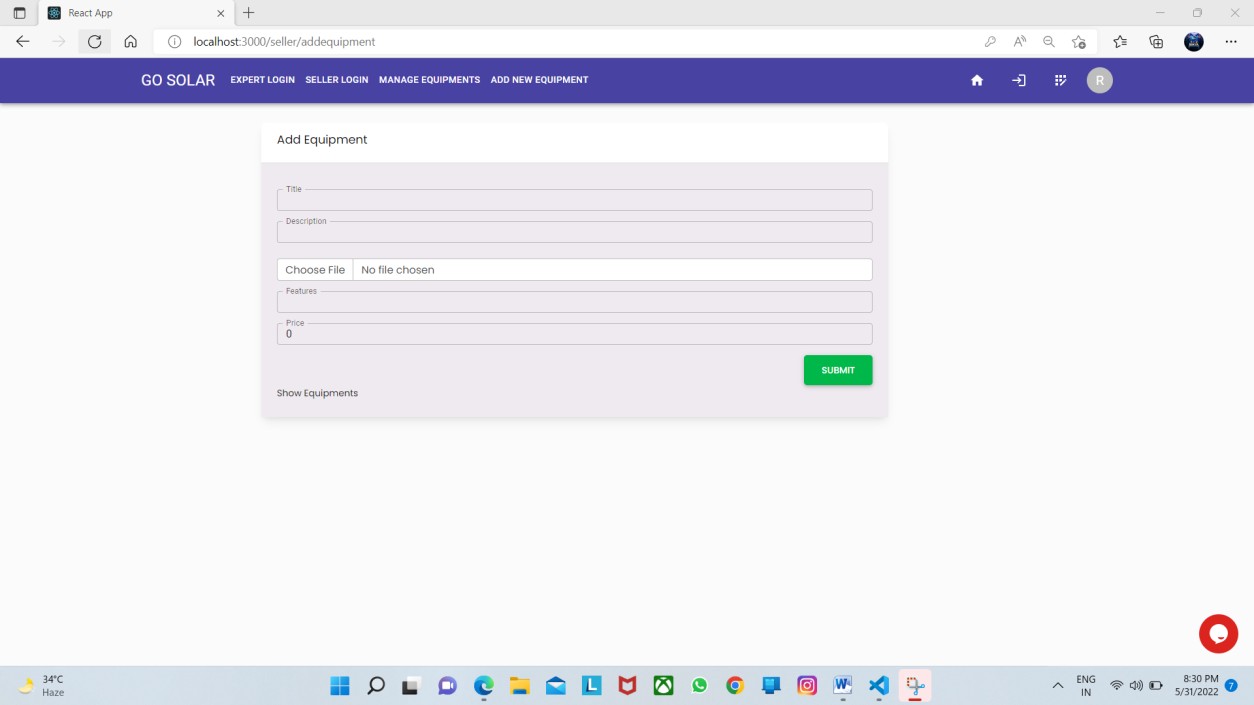


**Login Page –**

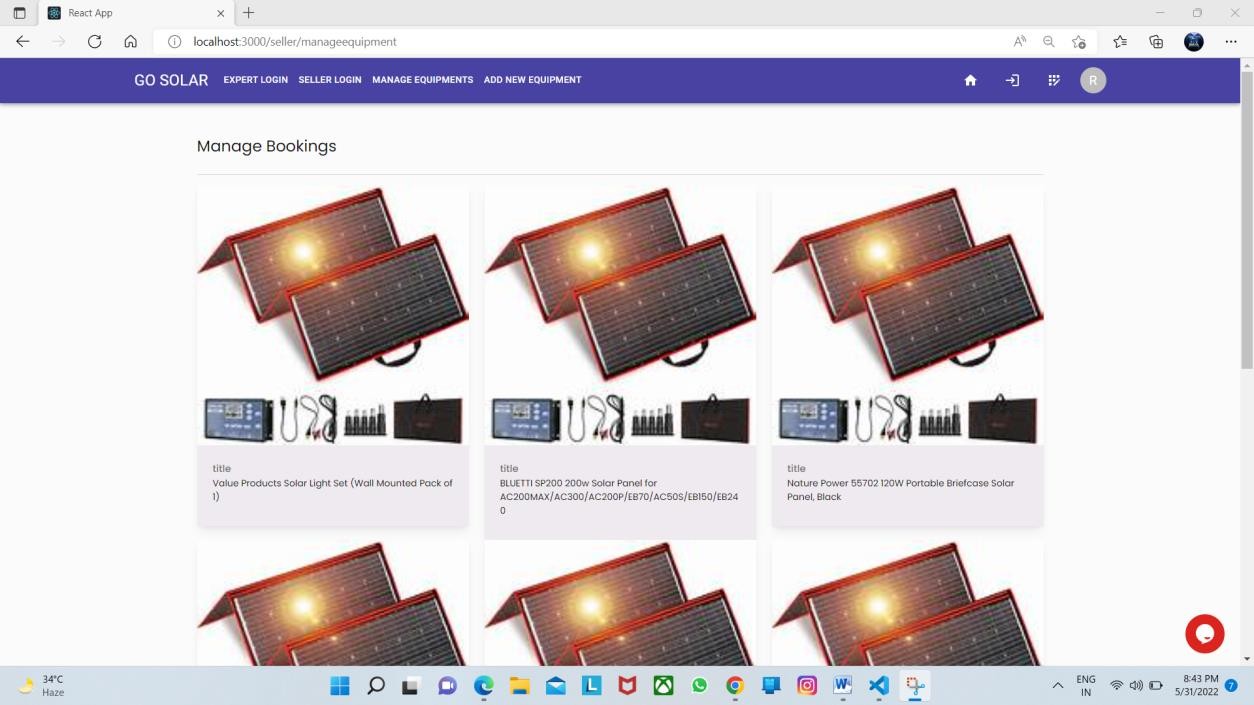


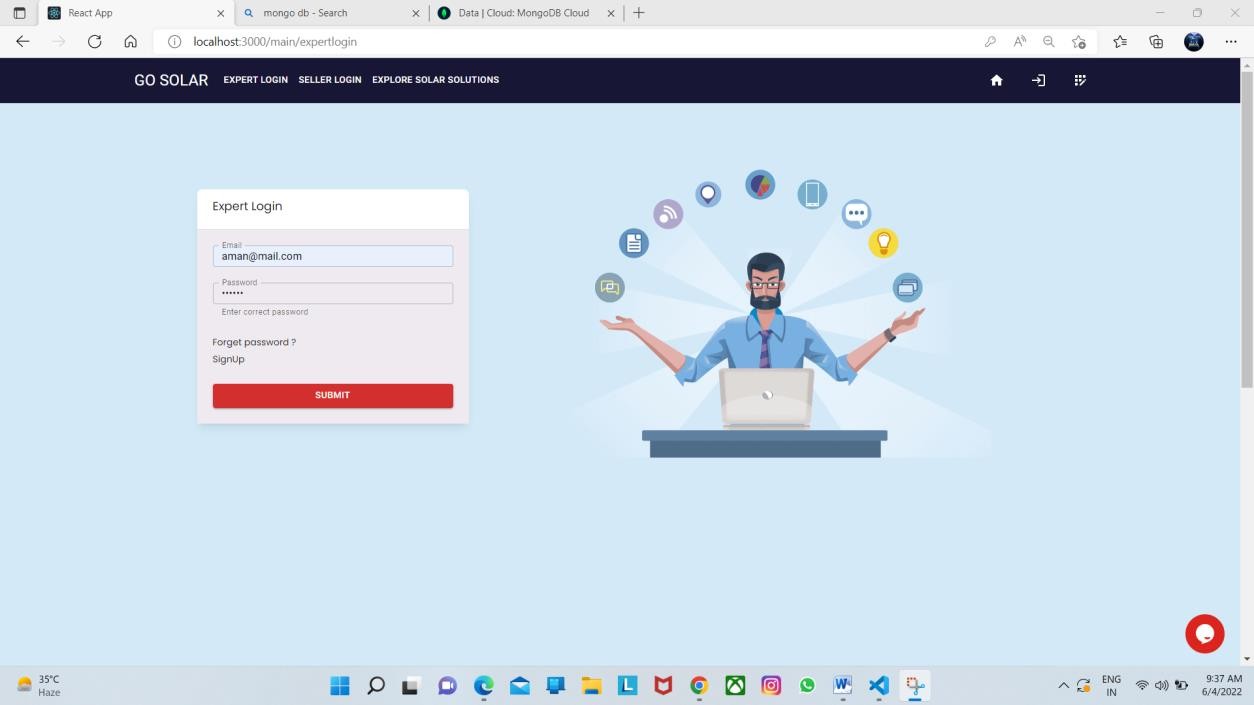
## Seller login page –

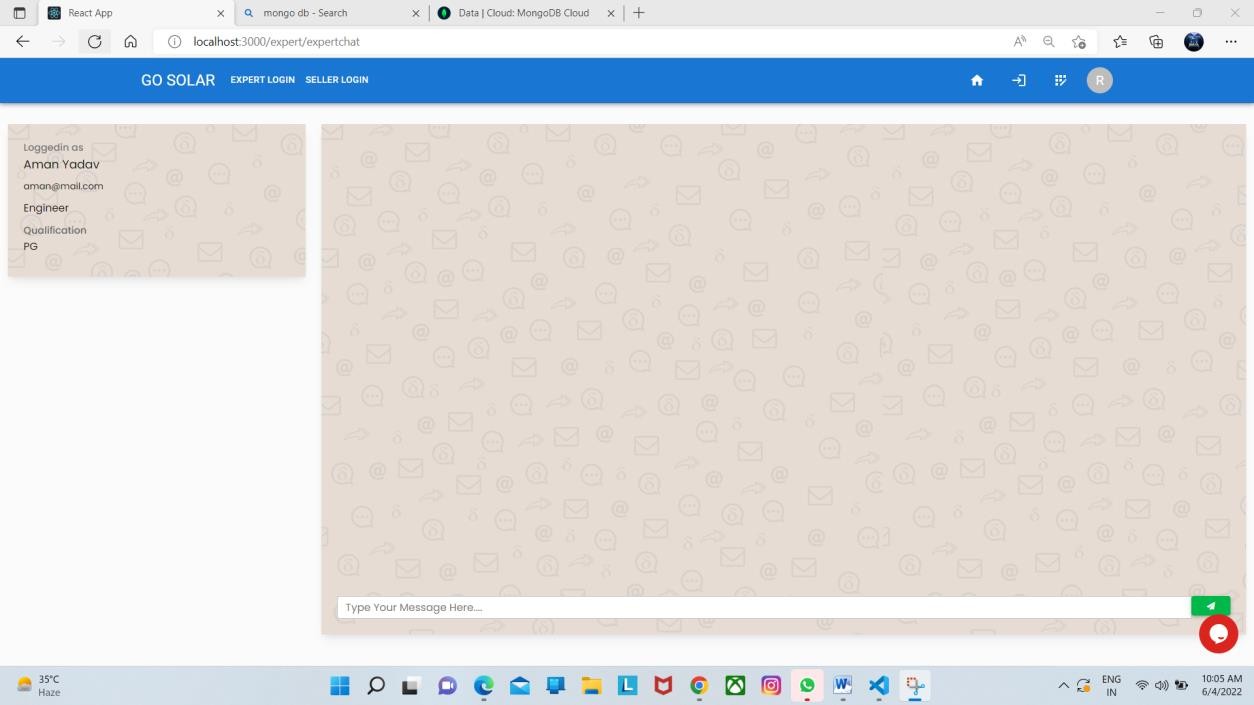




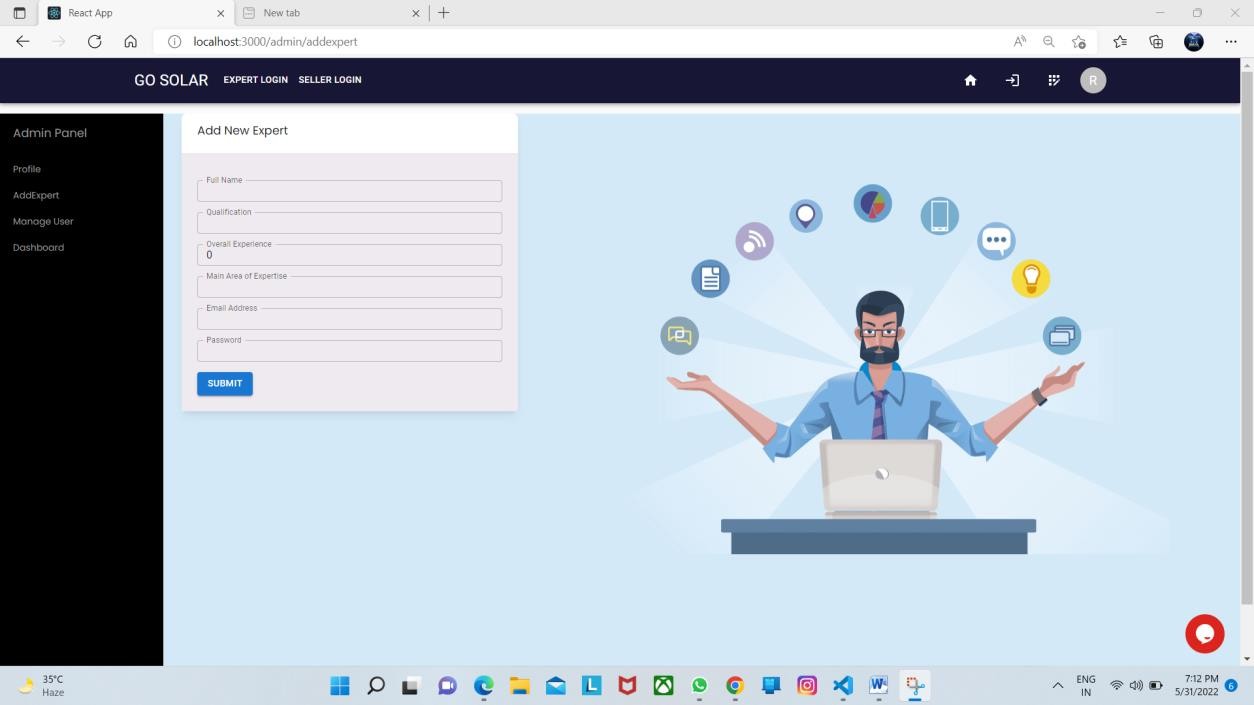
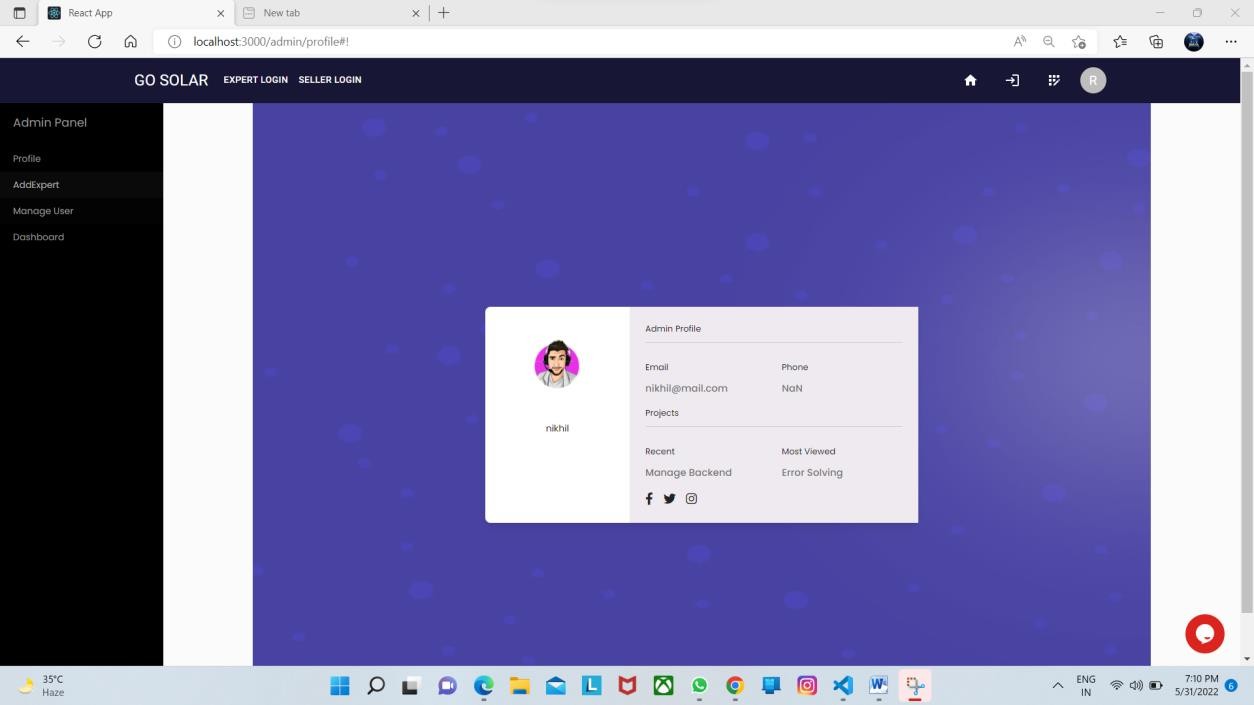
**Seller can manage equipment-**

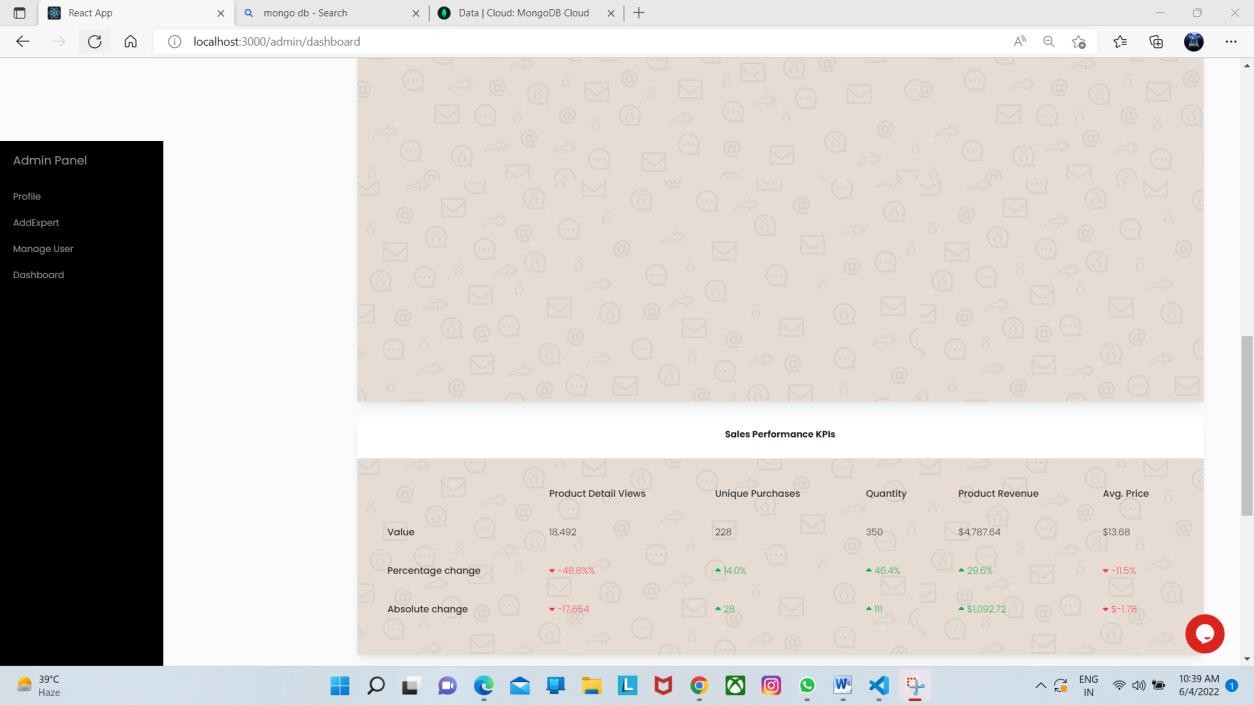




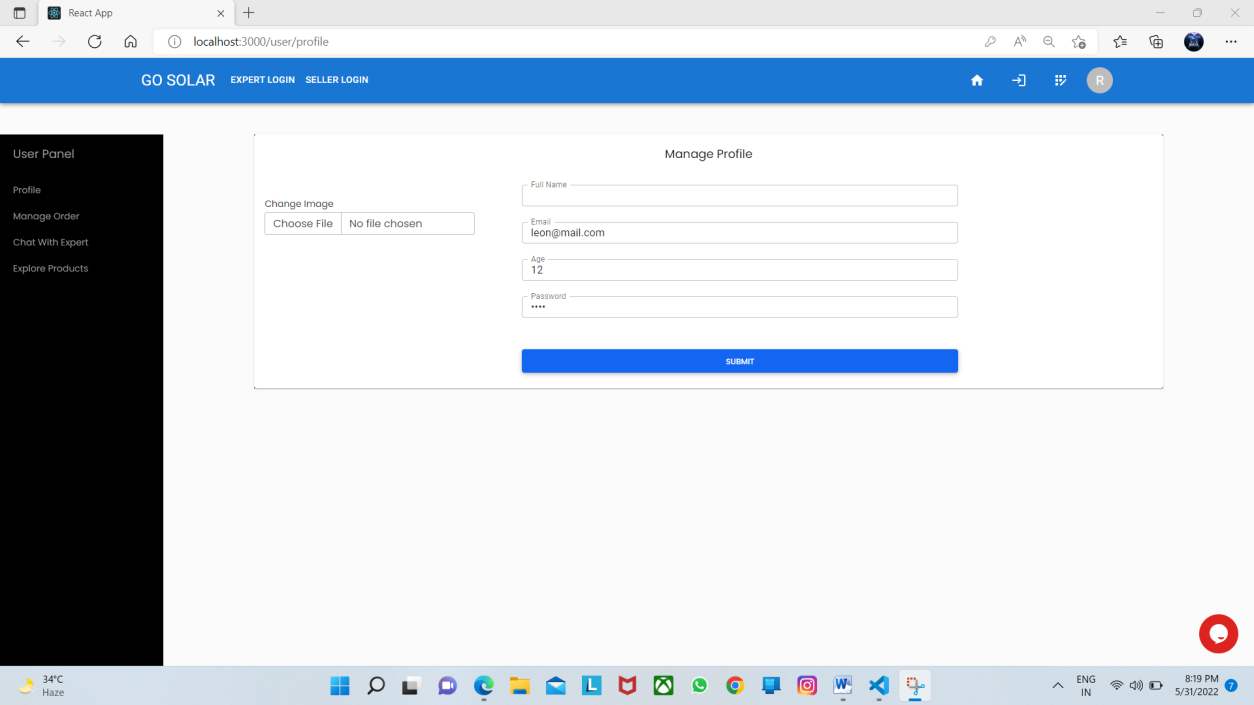


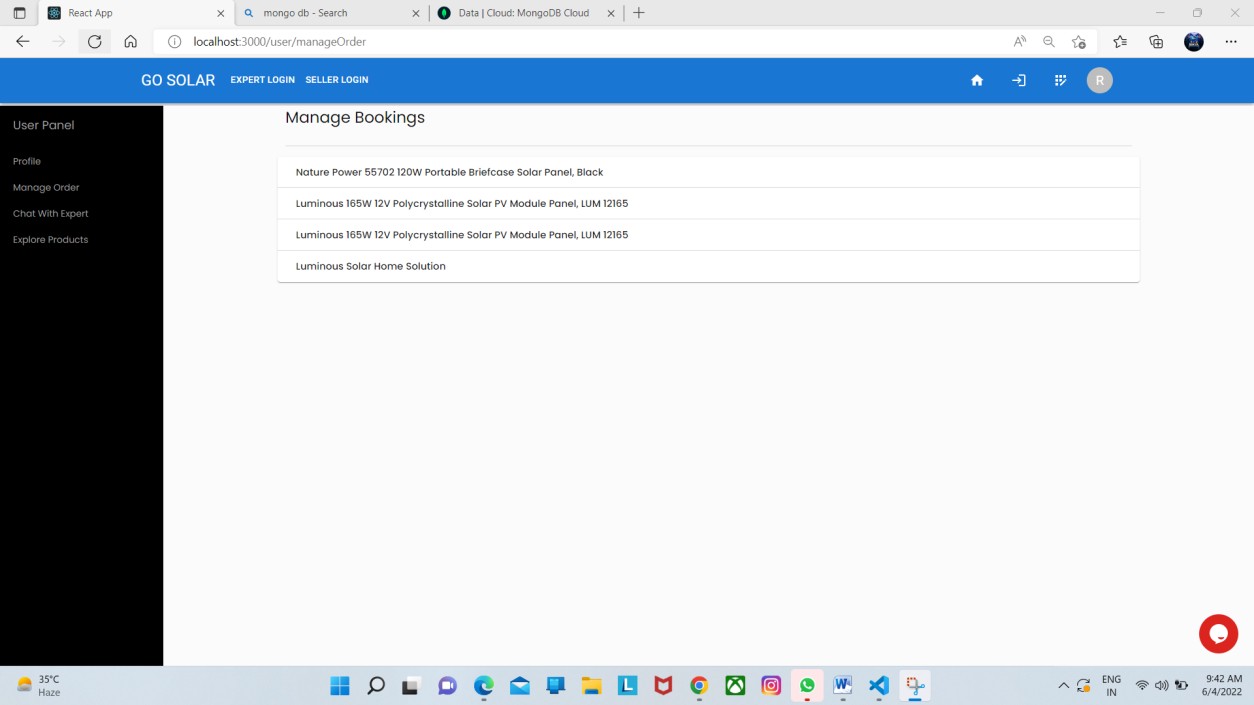
## Admin Dashboard –



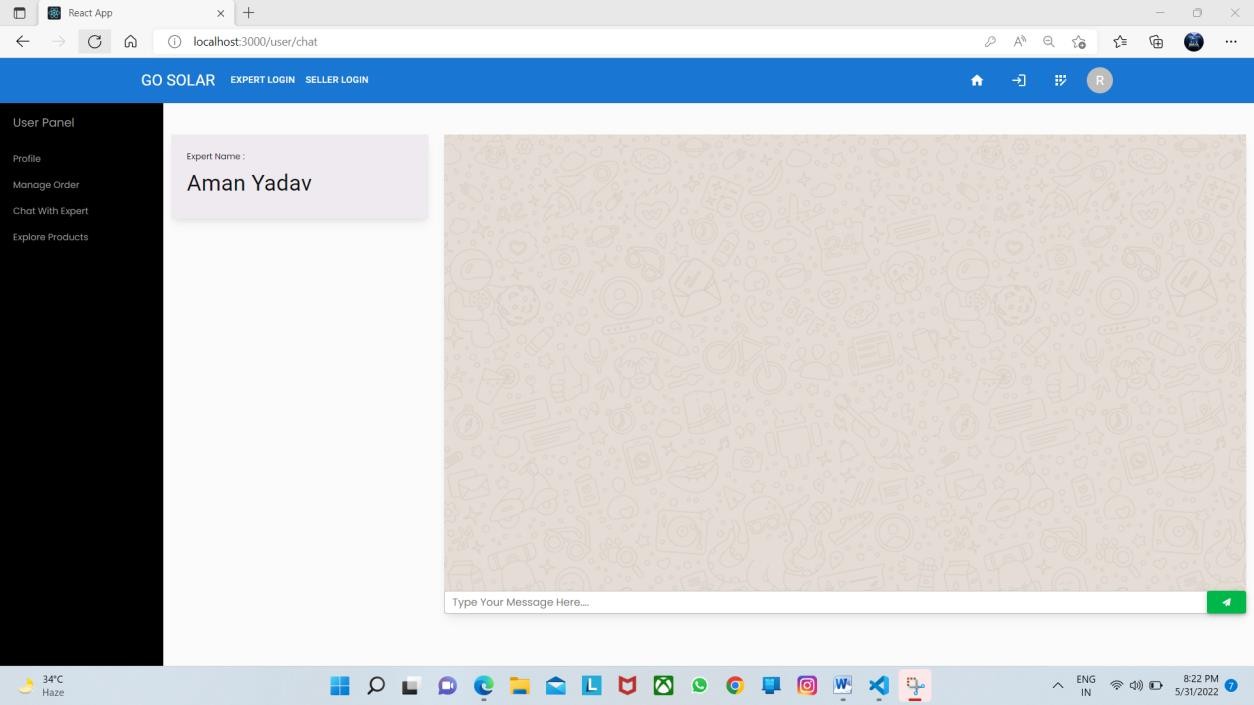


**User dashboard –**

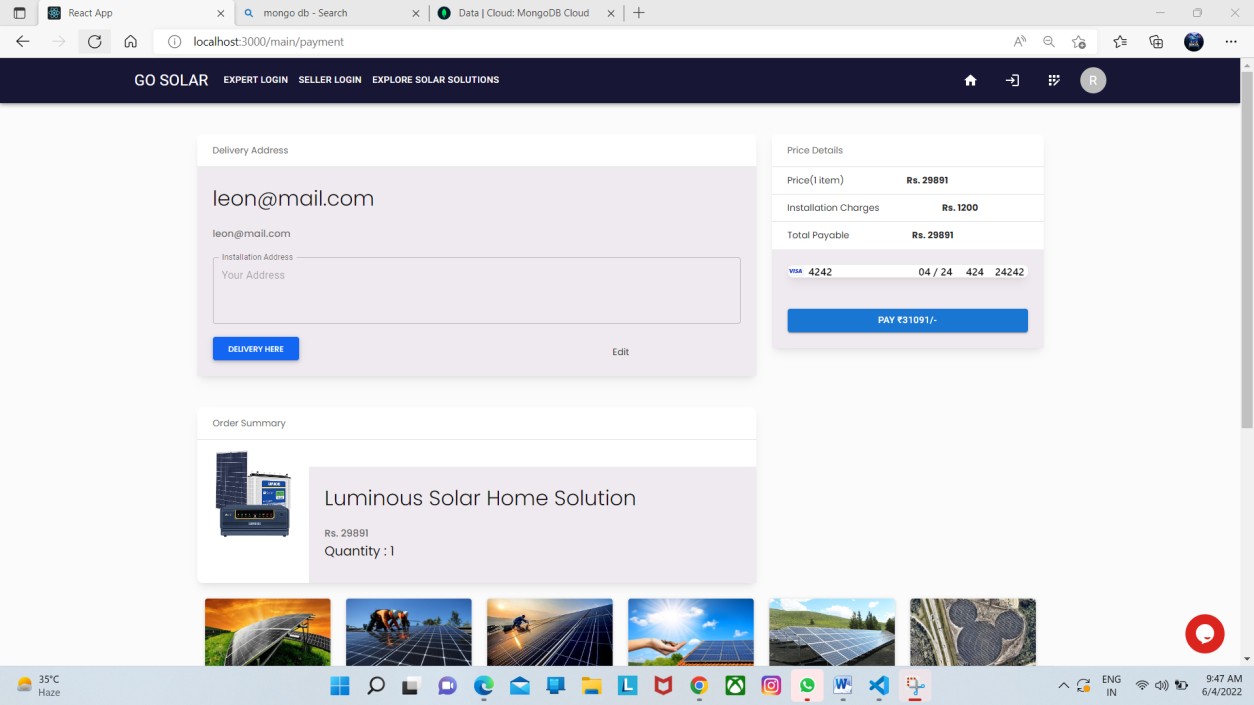


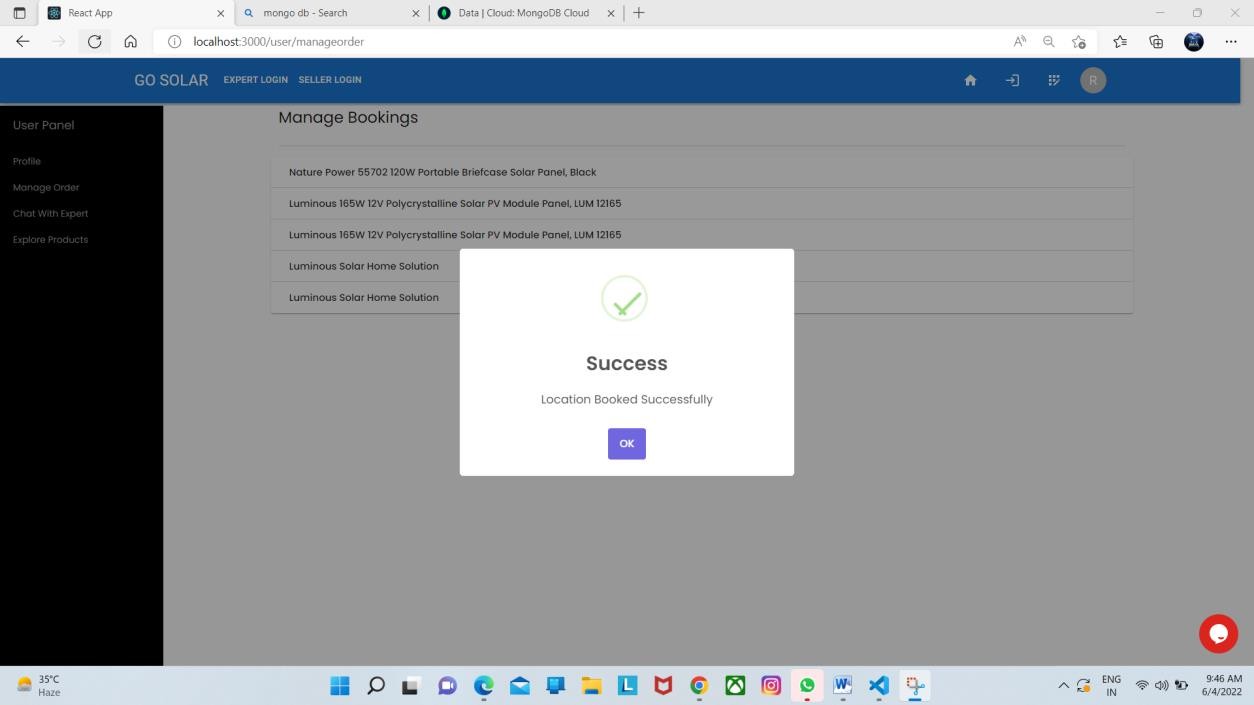


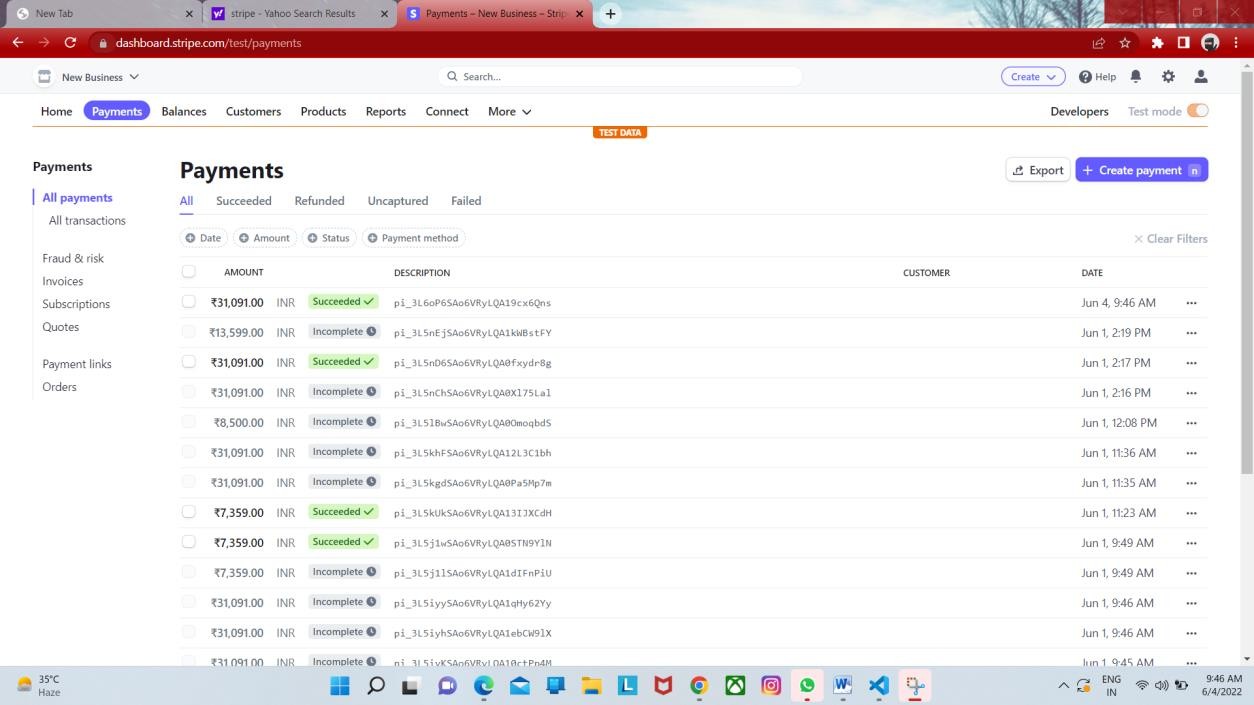
## Chat with expert -



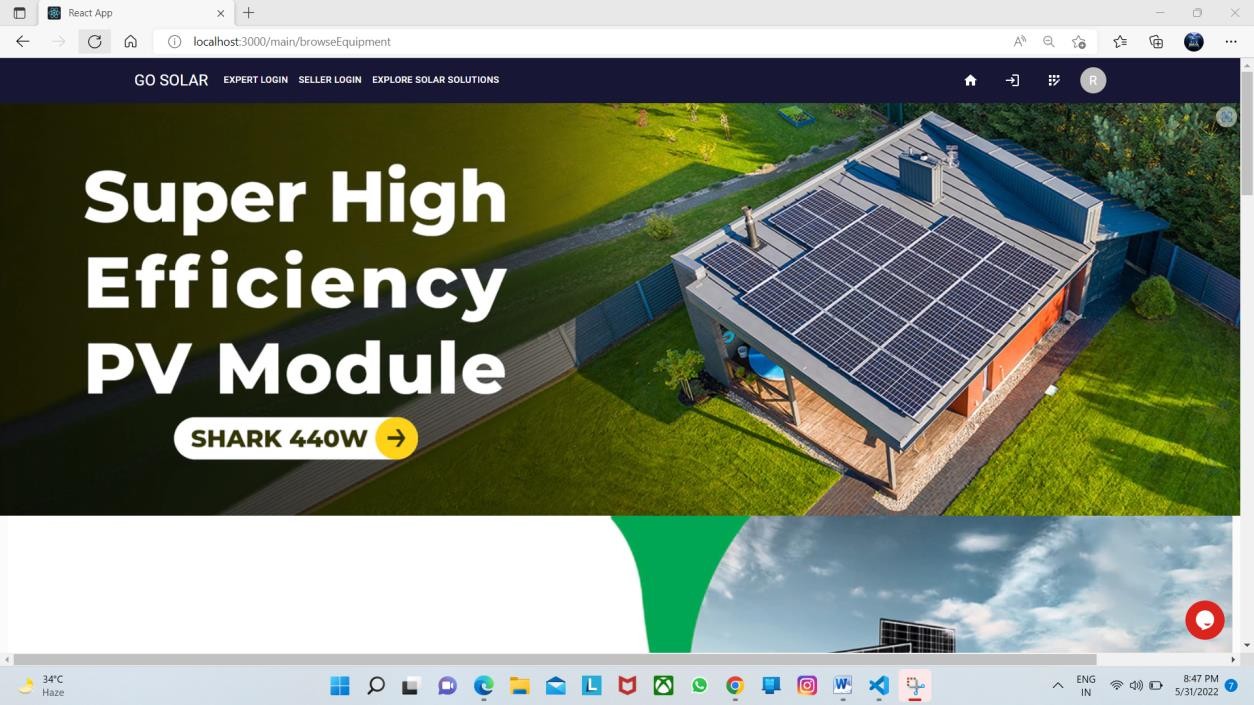
**Payment page –**

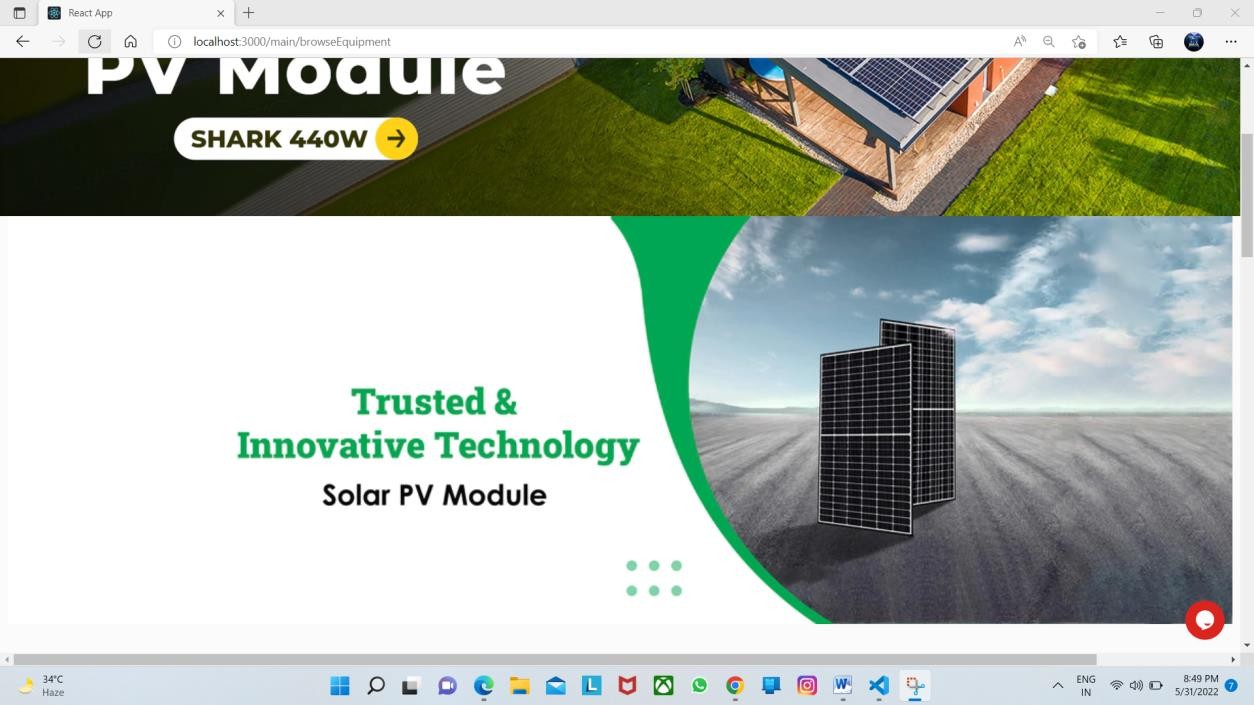


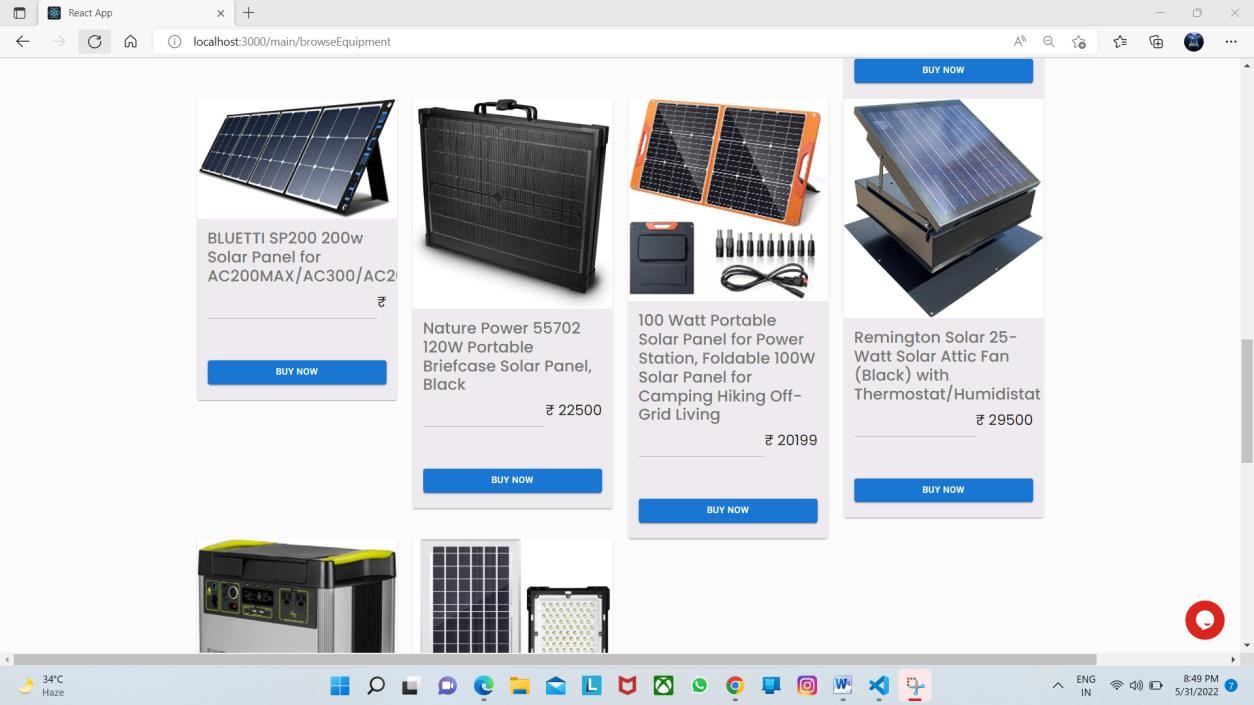


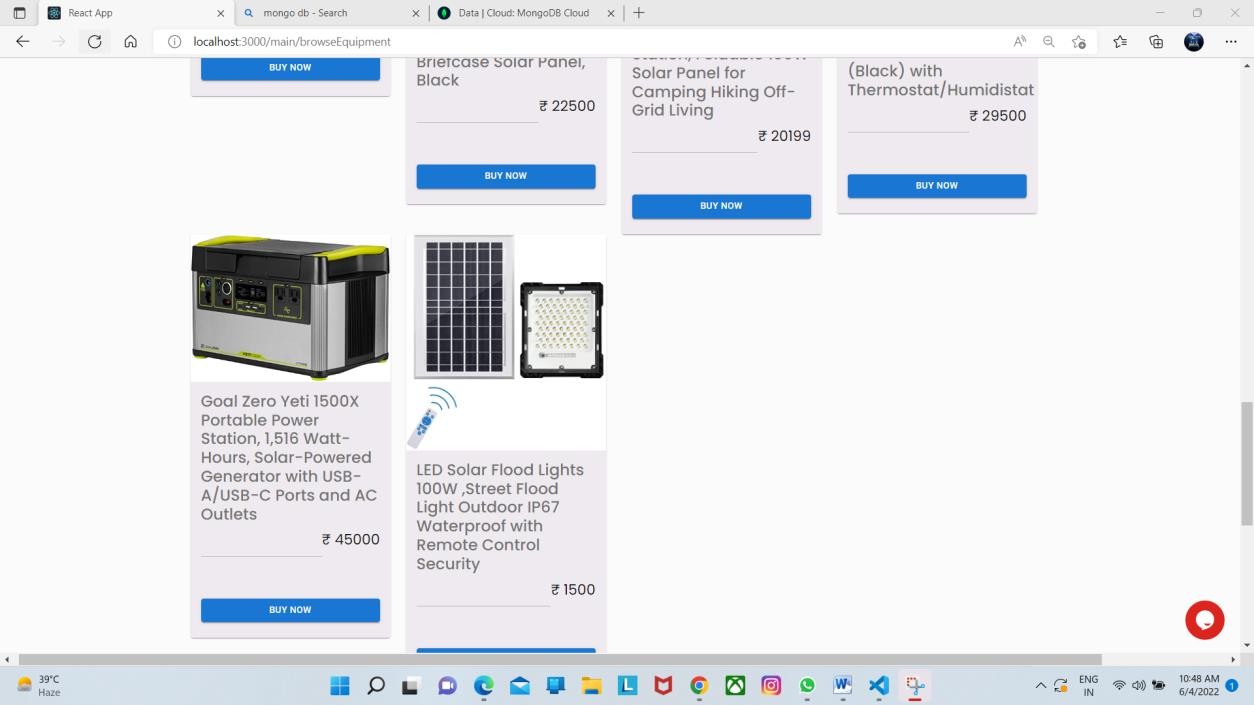


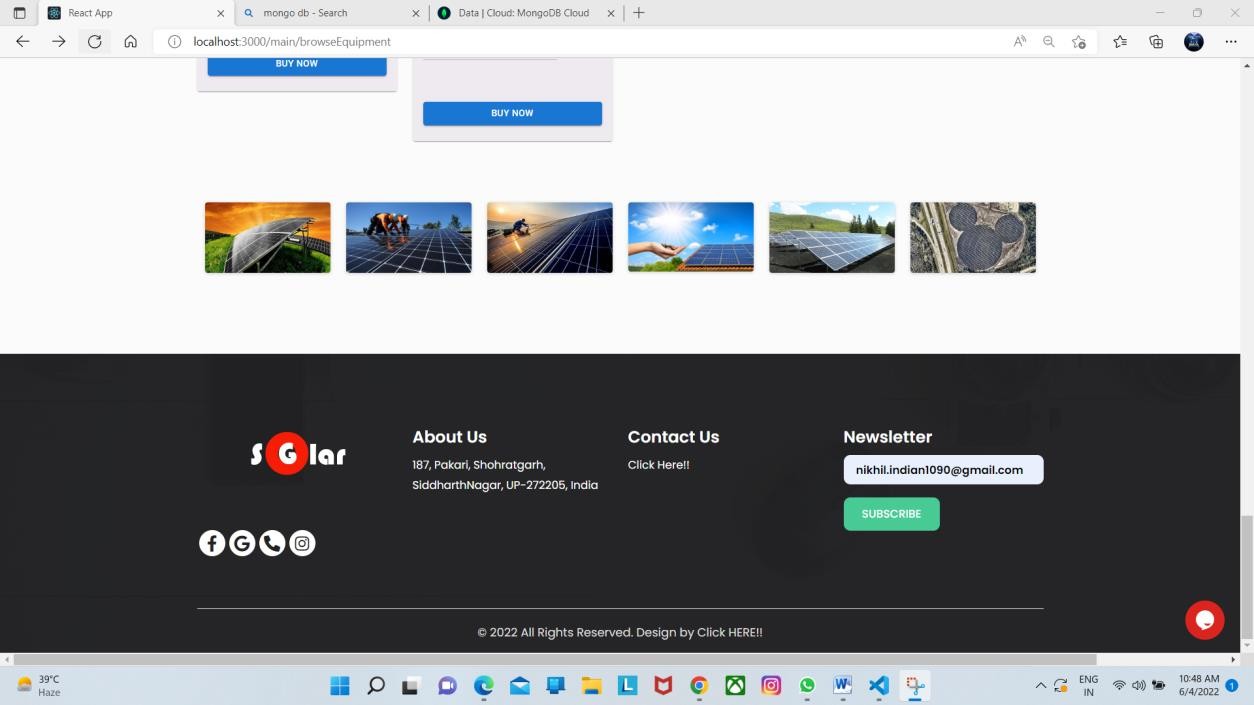
## Solar solution page –



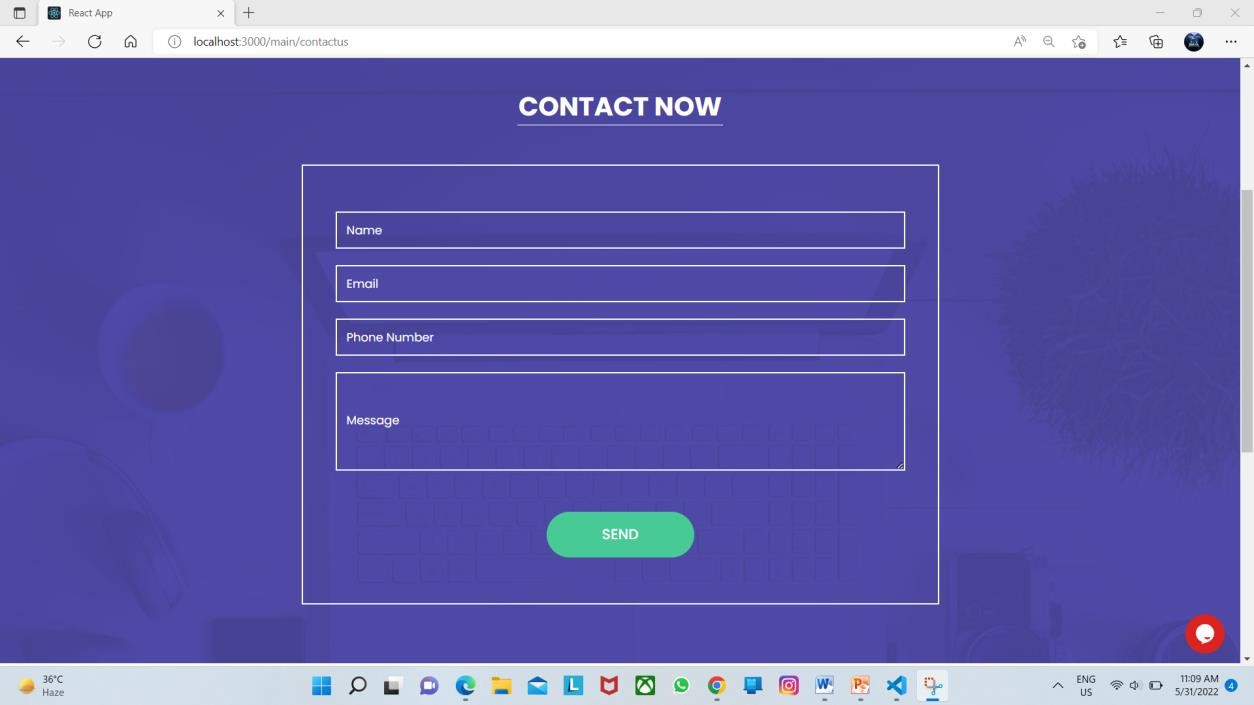








**Contact us page –**



|  |  |
| --- | --- |
| **Conclusion**   * In conclusion, the website plugin generator is an essential tool for anyone looking to enhance their website's functionality and user experience. With its user-friendly interface and customizable options, it allows users to create a plugin that perfectly fits their specific needs, without the need for extensive coding knowledge.   **Bibliography**  **Web Reverences:-**   * + https[://w](http://www.google.com/)ww.[google.com/](http://www.google.com/)   + [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)   + [**https://www.tutorialspoint.com/index.htm**](https://www.tutorialspoint.com/index.htm)   + **https://en.wikipedia.org/wiki/Main\_Page** |  |
| 61 |