|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** |  | **Subject** | **Page Number** |
| **1** |  | **Project Profile & Company Profile** | **2** |
| **2** |  | **Introduction to tools** | **5** |
| **3** |  | **System Study** | **16** |
|  |  | Existing System | **16** |
|  |  | Proposed System | **16** |
|  |  | Scope of the Proposed System | **17** |
|  |  | Aim and Objective of the Proposed System | **18** |
|  |  | Feasibility Study | **18** |
|  |  | Operational Feasibility | **18** |
|  |  | Technical Feasibility | **18** |
|  |  | Economic Feasibility | **18** |
| **4** |  | **System Analysis** | **19** |
|  |  | Requirements Specification (along with System Modules) | **20** |
|  |  | Use Case Diagram | **21** |
|  |  | Activity Diagram | **22** |
|  |  | Class Diagram |  |
|  |  | *OR* |  |
|  |  | System Flowchart  Functional Decomposition Diagram  DFD   * Context Level DFD * First Level, Second Level …. | **23** |
| **5** |  | **System Design** | **27** |
|  |  | Data Dictionary | **28** |
|  |  | Screen Layouts | **30** |
|  |  | Reports |  |
| **6** |  | **System Testing** | **40** |
|  |  | Testing Strategies | **40** |
|  |  | Test Cases | **40** |
| **7** |  | Future Enhancement | **41** |
| **8** |  | Bibliography/References | **42** |

**Table of Contents**

1 PROJECT PROFILE

## Project Description:

The “Prevention of security breaches caused by SQL Injections”

has been used to develop this web application to override the problems prevailing in the Database based applications. This application is supported to eliminate and in some cases reduce the hardship caused SQL Injections. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

This application is reduced as much as possible to avoid the most common types of security breaches caused by SQL Injection that occurs in web applications that can result in serious data breaches, including theft of sensitive information and unauthorized access to systems.

This web application, which made for a solar panel company named Priyasha, as described above, can lead to secure, reliable and error free system using parameterized queries, input validations, limiting user privileges, regularly updating, firewalls and intrusion detection systems and educate the user.

Every organization, whether small or big, has challenges to overcome and manage the web applications and security, therefore we designed this exclusive web application with reducing the risks of SQL Injection attacks.

## Project Features:

* Features:
* Collaboration Of Various Marketing Tools
* Web-based Business Support
* Brand Integration
* Online Shopping
* Self-Registration for Clients
* Responsive Design Features
* Accreditation Support
* Natural User Interface
* Availability Features
* Admin/User side Features:
* Signing Up for placing order
* Reporting and Data Analysis
* Assessment Management & Live Feedback
* Quality Content
* Quick User Integration
* Easy Payment Methods
* Team Information Management

## Project Profile:

|  |  |
| --- | --- |
| Project Name: | Prevention of security breaches caused by SQL Injections |
| Front-End: | CSS, JavaScript, HTML |
| Back-End: | MySQL, PHP |
| Browsers: | Google Chrome, Mozilla, Safari |
| Platform: | Windows 10, 11 |
| Documentation Tool: | Microsoft Office Word 2019 |
| Guide: | Ms. Konica Soni |
| Submitted To: | Charotar University of Science and Technology |
| Developed By: | Prapti Jigneshbhai Patel |
|  | Atmik Maheshbhai Virani |

1. Introduction **to Tools**

# Hardware and Software Requirements:

|  |  |
| --- | --- |
| Hardware Specification | |
| Development Time: | **4 GB RAM, intel core i3 or Higher processor** |
| Run Time: | **1 GB RAM, intel core i3 or Higher processor** |

|  |  |
| --- | --- |
| Software Specification | |
| Browser: | Google Chrome, Mozilla, Safari |
| Operating System: | Windows 10, 11 |
| Front-End: | CSS, JavaScript, HTML |
| Back-End: | MySQL, PHP |
| Other Tools: | Microsoft Visual Studio Code |

# 2.2 Technology Used:

## 2.2.1 Back-End Tools:

* **MySQL Technology:**
* **Introduction to MySQL:**
* **MySQL** is an open-source relational database management system(RDBMS).
* It is the most popular database system used in PHP.
* **MySQL** is very fast, reliable, and easy to use database system. It uses standard SQL.
* **MySQL**  works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* **MySQL** is very friendly to PHP, the most appreciated language for web developments.
* **MySQL** supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this to a theoretical limit of 8 million terabytes.
* **Features of MySQL:**

1. **Scalability & Flexibility:**

With MySQL you can run deeply embedded applications and create data warehouses holding a humongous amount of data.

1. **High Performance & Availability:**

Provides fast load utilities with distinct memory caches and table index partitioning. MySQL can run high-speed master/slave replication configurations and its offers cluster servers.

1. **Data Types:**

Supports multiple data types like primitive, structured, documents, etc.

1. **It is secure:**

MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

1. **High Performance:**

MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.

1. **Dual Password Support:**

MySQL version 8.0 provides support for dual passwords: one is the current password, and another is a secondary password, which allows us to transition to the new password.

1. **Client/Server Architecture:**

MySQL follows working of a client/server architecture. There is a database server(MySQL) and arbitrarily many clients (application programs), which communicates with the server; that is, they can query data, save changes, etc.

* **PHP:**
* **Introduction to PHP:**
* **PHP** started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
* **PHP** is a recursive acronym for “PHP: Hypertext Pre-processor”.
* **PHP** is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* **PHP** supports a large number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* **PHP** supports large number of major protocols such as POP3, IMAP, LDAP. PHP4 added support for java and distributed object architectures(COM and COBRA), making n-tier development a possibility for the first time.
* **PHP** is forgiving: PHP language tries to be as forgiving as possible.
* **PHP** Syntax is C-like.
* **Advantages of PHP:**

**PHP** has remained one of the most versatile and pragmatic web development languages in the world today.

**PHP** range of functionalities, amazing array of add-ins to extend functionalities, its open-source nature, and tremendous online community support has made PHP a perennial favourite amongst newbies as well as established development agencies worldwide.

* **Easy and Simple to Learn**
* **Extremely Flexible**
* **Easy Integration and Compatibility**
* **Efficient Performance**
* **Cost-Efficient**
* **Gives Web Developer More Control**

## 2.2.2 Front-End Tools

* **CSS:**
* **Introduction to CSS:**
* **CSS (Cascading Style Sheets)** is a language for creating style sheets that define how a web page or web application looks visually.
* The front-end of a web programme can be made rich and dynamic by combining **CSS** with HTML and JavaScript.
* **CSS** offers a collection of guidelines that can be used to alter the color, font, layout, and other stylistic components of HTML elements.
* **CSS** is made to be extremely versatile and customizable.
* **Features of CSS:**

1. **Selectors:**

To target particular HTML components and apply styles to them, **CSS** makes use of selectors.

1. **Properties:**

A variety of **CSS** attributes, including color, typeface, size, padding, margin, and others, can be used to modify how HTML components look.

1. **Values:**

Color names, hex symbols, numeric figures, and more can all be used as variables for **CSS** attributes.

1. **Layout:**

To manage the positioning and arrangement of HTML components, **CSS** offers layout features like positioning, floats, and flexbox.

1. **Responsive Design:**

In order to build flexible designs that change according to various screen widths and devices, **CSS** contains features like media queries and viewport units.

* **JavaScript:**
* **Introduction to JavaScript:**
* It is used to build dynamic and engaging user interfaces. **JavaScript** is a high-level computer language that is mainly used for front-end web development.
* Brendan Eich created **JavaScript** for the first time at Netscape in 1995.
* **JavaScript** is an interpreted language, which means that there is no need for compilation because the code is run by the browser as it is met.
* • **JavaScript** is used to add utility and engagement to web sites, including form validation, dynamic content, and engaging elements like sliders and animations.
* • Back-end programming, including server-side coding with Node.js, can also be done in **JavaScript**.
* **Features of JavaScript:**

1. **Variable and Datatypes:**

Numerous datatypes, such as integers, strings, Booleans, objects, and groups, are supported by **JavaScript**.

1. **Operators:**

Numerous operators are available in **JavaScript**, such as math, comparison, and logical operators.

1. **Control Structures:**

To manage the processing of programs, **JavaScript** has control structures like loops, switch statements, and if/else statements.

1. **Functions:**

**JavaScript** gives programmers the ability to create repeatable methods that can be used throughout the code.

1. **DOM Manipulation:**

**JavaScript** gives developers access to a web page's Document Object Model (DOM), enabling them to change the page's layout and content in reaction to user input.

1. **Events:**

The various event handlers offered by **JavaScript** enable developers to react to user activities like clicks, scrolls, and form entries.

* **HTML**
* **Introduction to HTML:**
* Web sites and web apps are made using the markup language **HTML (Hypertext Markup Language)**.
* **HTML** offers a method to organise and organise material, including text, images, and multimedia, on the web.
* In order to create a dynamic and engaging user experience, **HTML** must be used in conjunction with other web tools like CSS and JavaScript.
* The layout and substance of online sites are defined by **HTML** using tags and attributes.
* **HTML** tags are used to designate up website components, while attributes give the parts more details.
* **Features of HTML:**

1. **Elements:**

**HTML** offers a broad variety of components that can be used to organise and arrange content on a web page, including headers, paragraphs, lists, pictures, and tables.

1. **Attributes:**

**HTML** components are capable of having attributes that reveal more details about them, such as the origin of a picture or the destination of a link.

1. **Forms:**

Users can input data and engage with the website by using interactive forms made with **HTML**.

1. **Semantic markups:**

With the introduction of semantic components like article, section, and nav, the material of a web website can now have more significance and structure.

1. **Accessibility:**

A variety of accessibility features in **HTML**, like alt text for pictures and ARIA properties for interactive components, increase the usability of online content for people with impairments.

# System Study

# Existing System and its Drawbacks:

* Web apps can be used for a variety of marketing initiatives, enabling companies to connect and interact with clients online.
* Web apps can be used for a wide range of marketing initiatives, giving businesses the opportunity to connect with and interact with consumers online while also boosting sales and profits. Businesses can increase their online visibility and successfully pinpoint their target population by utilising the power of web apps.
* **Disadvantages of System:**
* Attacks using SQL injections can lead to security risks, such as data breaches where private data may be taken, changed, or removed by unauthorised people.
* Data leaks may result in monetary losses, harm to one's image, and legal repercussions.
* Users' confidence in online applications may be harmed by successful attacks, possibly costing businesses customers and income.
* Data loss and corruption may result from breaches that jeopardise the database security of an online service.
* If an online programme has many lines of code and has been in use for a long time, fixing the security flaws may take a long time and be expensive.

# Proposed System:

Electronic based marketing is known as E-commerce. A customer can easily shop or look out the products from their home. E-commerce has a comprehensive range of marketing strategies and technologies; from online searching orders to placing orders to online payment, making transactions cashless. E-commerce made the tracking of the orders easy and just up to the hand of the customer.

## Scope of Proposed System:

Electronic based marketing is known as E-commerce. This web application is designed and developed without SQL Injection vulnerabilities.

* A safer online programme would be less susceptible to theft and data leaks.
* Sensitive data would be kept and recovered without the threat of unauthorised access or change, making the database more durable and dependable.
* User inputs wouldn't ever be immediately concatenated into the SQL query text because parameterized queries would be used.
* Before user inputs are used in database queries, input validation would be applied to make sure they are in the anticipated style and range, stopping users from inserting malicious code or characters into the database.
* To fix any known flaws, routine software and database management system upgrades are made.
* Easy to operate and have a good user and operator.
* It satisfies the user requirement.
* Easy to understand for the user and operator.
* Be expandable
* Delivered on schedule within the budget.

## Aim and Objectives of Proposed System:

## Project Modules:

* Admin
* User

## Modules Vice Objectives:

* **Admin:**
* Admin manage user in system and maintain their information.
* Admin can manage his profile and able to change the password.
* Admin can manage payroll system and generate monthly salary slip.
* Admin can also generate offer/coupons for new user.
* Admin can review and manage all user reports.
* Admin can add category and sub-category for the products.
* **User:**
* User can register/login in web application themselves.
* User can view product according to the categories they prefer.
* User can manage his profile and can also change the password.
* User can contact the admin online on web application.
* User can add products to cart.
* User can purchase the selected product from the web application.

## Feasibility Study:

* The modern era constantly introduces new information and communication tools. Even in terms of promotion, they are developing. E-commerce's significance in the current industry is still expanding. It is put into place to help the marketing process and boost its effectiveness and quality. E-commerce is used to supplement conventional marketing strategies or is progressively taking their place. Market purchase possibilities are evolving and becoming more contemporary.
* Economic Feasibility:

In developing countries, however, a complete web implementation of e-commerce is eventually possible and will have more advantages for both the customer and the seller. It was created with cost-effective design and development, and income sources should be varied for security.

* Operational Feasibility:

Operational viability is a metric used to assess how effectively a planned system addresses the issue, seizes opportunities found during scope definition, and complies with requirements found during the requirements analysis stage of system development.

* Technical Feasibility:

The specifics of how you plan to provide the goods or services to consumers are evaluated by a technical feasibility study. It concerns the supplies, labour, essential transit, the location of your company, and the technology required to make all of this happen. The logistics or operational plan for how your company will create, store, transport, and monitor its goods or services is known as the "production plan."

# System Analysis

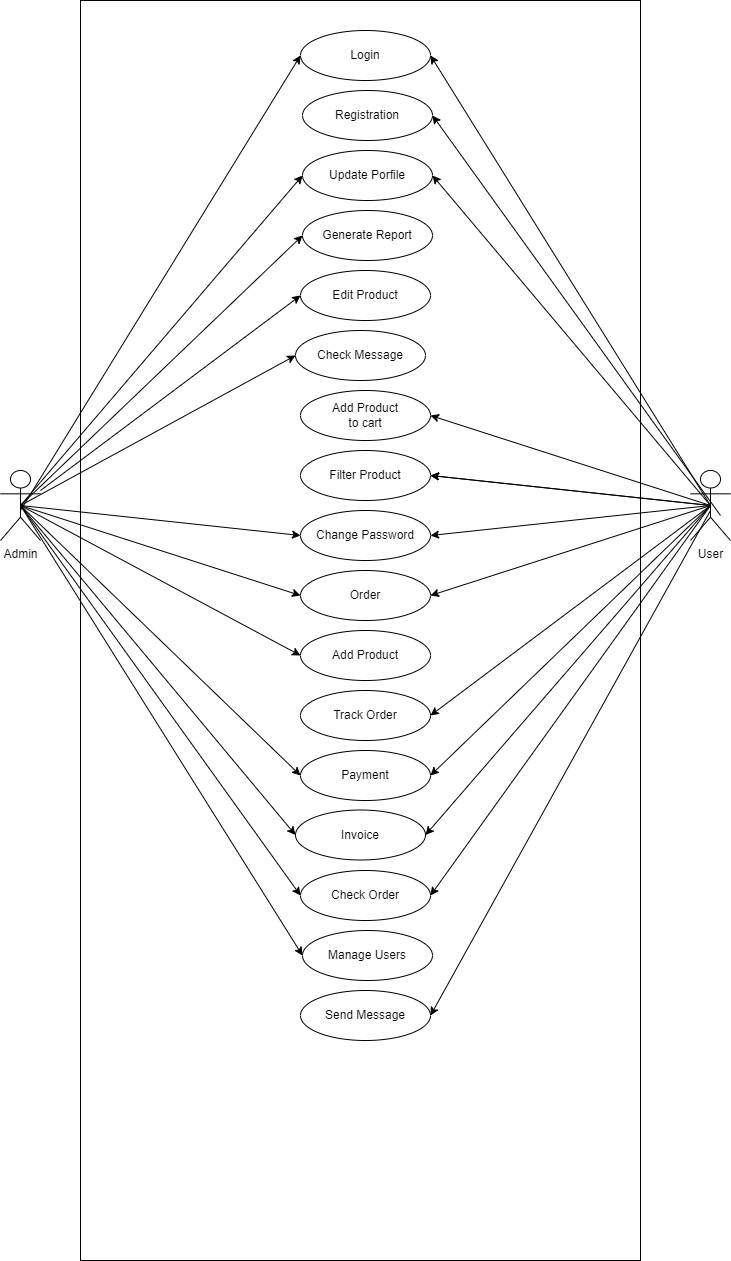
## Requirement Gathering and Analysis

* The first step in web application for solar panels without the risk of SQL Injections and it’s breaches is touted as the most complex and the most layered process. It involves interactions, discussions and data gathering and sets the base for the Solar Panel web application. But, quite often it ends up becoming a to and from task wherein there is always a new list of unanswered questions as the marketing gets started.
* Very often this thread of questions continues right till the end, forcing last minutes additions, changes, and modifications that slowdowns the development process, affects the quality and impacts the project delivery dates too.
* It involves identifying stakeholders, conducting interviews, developing use cases, documenting requirements, prioritizing them, validating them with the stakeholders, and reviewing them with development team. This process helps ensure the web application meets stakeholder needs and objectives, and is designed to function effectively and efficiently within the project timeline and budget.
* **Understanding the process:**
* Why do you need it?
* What problems will it solve?
* What do client need to know after they’ve selected the product?
* What do staff need to differently?
* What would happen if client does not checkout?
* How will the client know if the order is placed?
* **Requirement Analysis:**

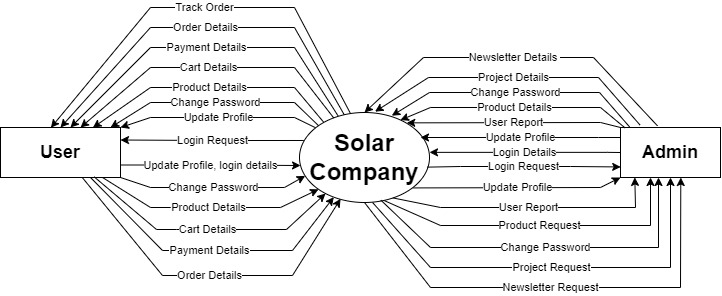
We analysis our gathered information and we have decided our system should have following functionalities.

* The system should have a register option.
* The user should be able to view details about the business, its goods, and services.
* The system must offer protection to data.
* The system must offer a secure registration option.
* Users ought to be able to purchase the goods they want.
* The system ought to accept online payments and produce bills with distinctive IDs.
* The user should be able to return the merchandise within a certain time frame if it is incompatible.
* The ability for users to submit and read evaluations and ratings.
* The administration must be able to control the staff.

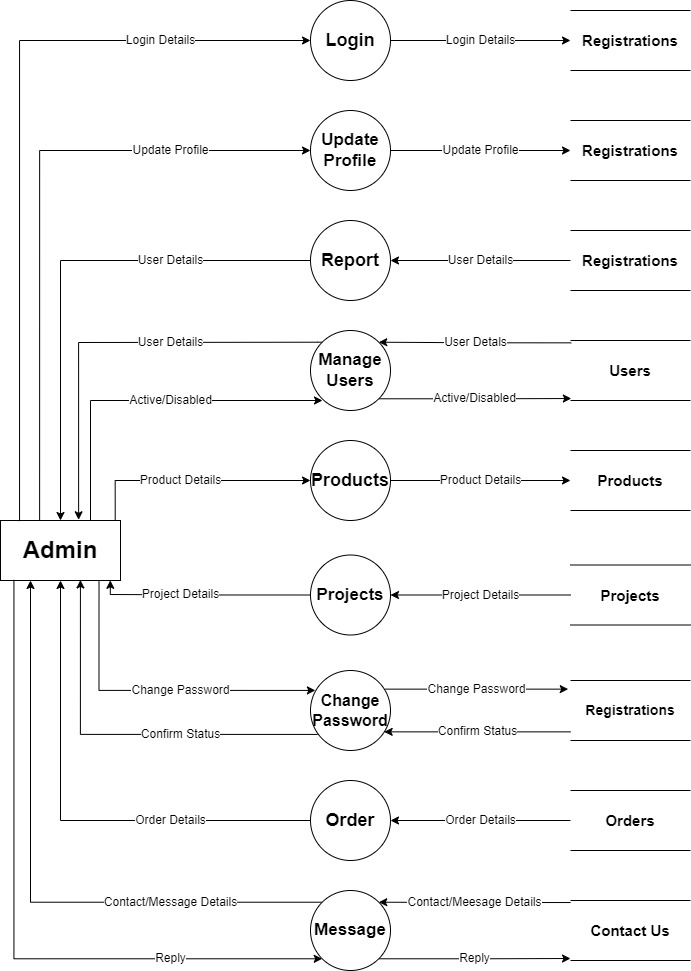
## Use Case Diagram

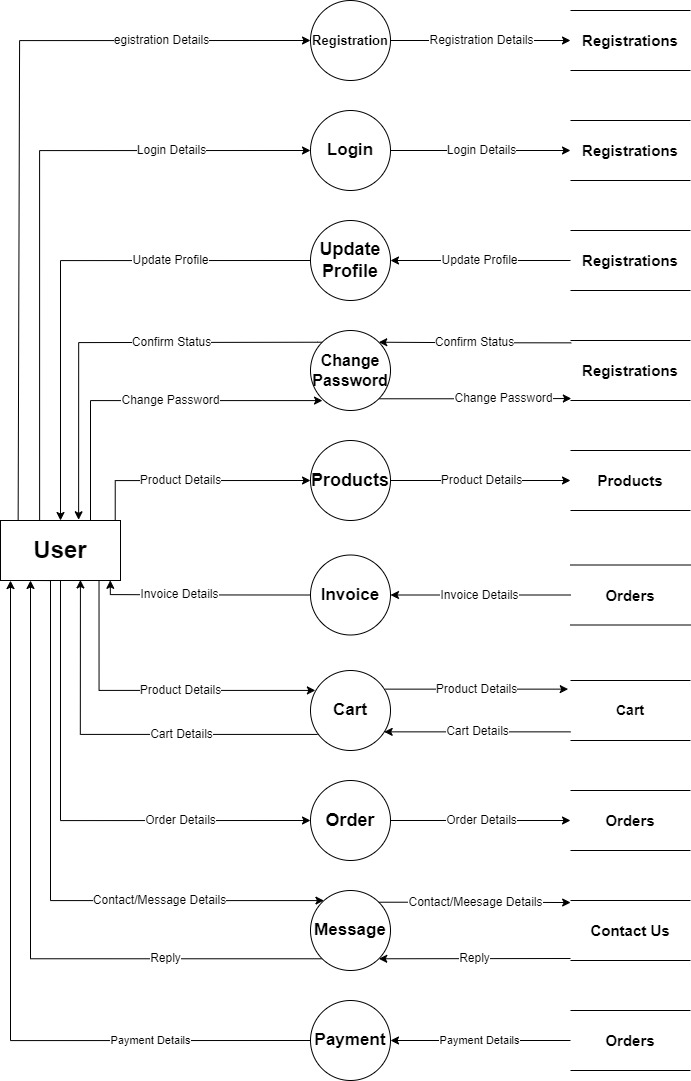


## Data Flow Diagram

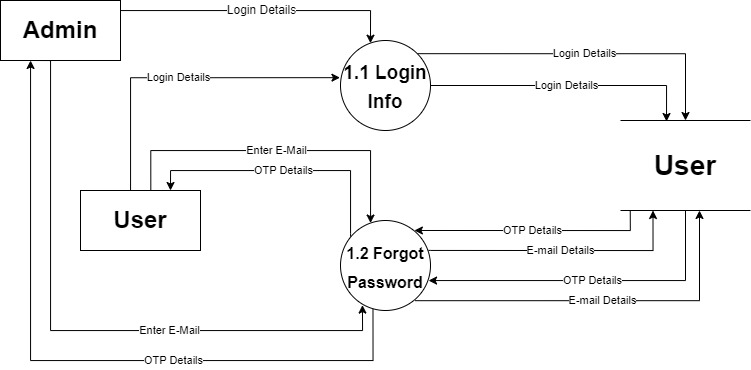


**[Context Level]**

**[First level of Admin]**

**[First level of User]**

**[Second Level DFD]**

****

# System Design

## Data Dictionary:

|  |  |
| --- | --- |
| Name: | Login Detail |
| Alias: | NULL |
| Where used?/ How used? | To login For User and Admin (Input/Output) |
| Content Description: | Email, Password |
| Supplementary Information: | Email or Password must be Unique.  Login Process takes detail and verify in database. |

|  |  |
| --- | --- |
| Name: | Update Profile |
| Alias: | NULL |
| Where used?/ How used? | To update data for User and Admin (Input) |
| Content Description: | Name, MobileNo., Email, Address, Pincode |
| Supplementary Information: | Email or MobileNo. must be Unique.  Update Profile Process takes detail and updates database. |

|  |  |
| --- | --- |
| Name: | User Detail |
| Alias: | NULL |
| Where used?/ How used? | To show User Detail in Admin (Output) |
| Content Description: | Name, MobileNo., Email, Address, Pincode |
| Supplementary Information: | Admin can see user data. |

|  |  |
| --- | --- |
| Name: | Product Detail |
| Alias: | NULL |
| Where used?/ How used? | Product (Input/Output) |
| Content Description: | Product id, Product name |
| Supplementary Information: | Product id must be Unique.  Admin can add Product. |

|  |  |
| --- | --- |
| Name: | Change Password |
| Alias: | NULL |
| Where used?/ How used? | Create New Password (Input) |
| Content Description: | Password |
| Supplementary Information: | Enter old and new password.  If old password is correct than the user can create new password or else it won’t allow. |

|  |  |
| --- | --- |
| Name: | Order Details |
| Alias: | NULL |
| Where used?/ How used? | Place Order or Order Status (Input/Output) |
| Content Description: | Product id, name, email, mobileno., address, pincode, payment id, amount, order id, order date |
| Supplementary Information: | Order id must be unique.  User can place the order.  Admin can see order status and details. |

|  |  |
| --- | --- |
| Name: | Registration Detail |
| Alias: | NULL |
| Where used?/ How used? | Registration For New User. (Input/Output) |
| Content Description: | Name, email, address, pincode, password, User\_id |
| Supplementary Information: | User\_id must be unique.  Only users can register themselves. |

|  |  |
| --- | --- |
| Name: | Cart Detail |
| Alias: | NULL |
| Where used?/ How used? | Product can be add to cart. (Input/Output) |
| Content Description: | Product id |
| Supplementary Information: | User can add product to cart. |

|  |  |
| --- | --- |
| Name: | Payment Detail |
| Alias: | NULL |
| Where used?/ How used? | Payment for Products (Input) |
| Content Description: | User id, product id, payment id |
| Supplementary Information: | Payment id is always unique.  User can pay online or cash on delivery. |

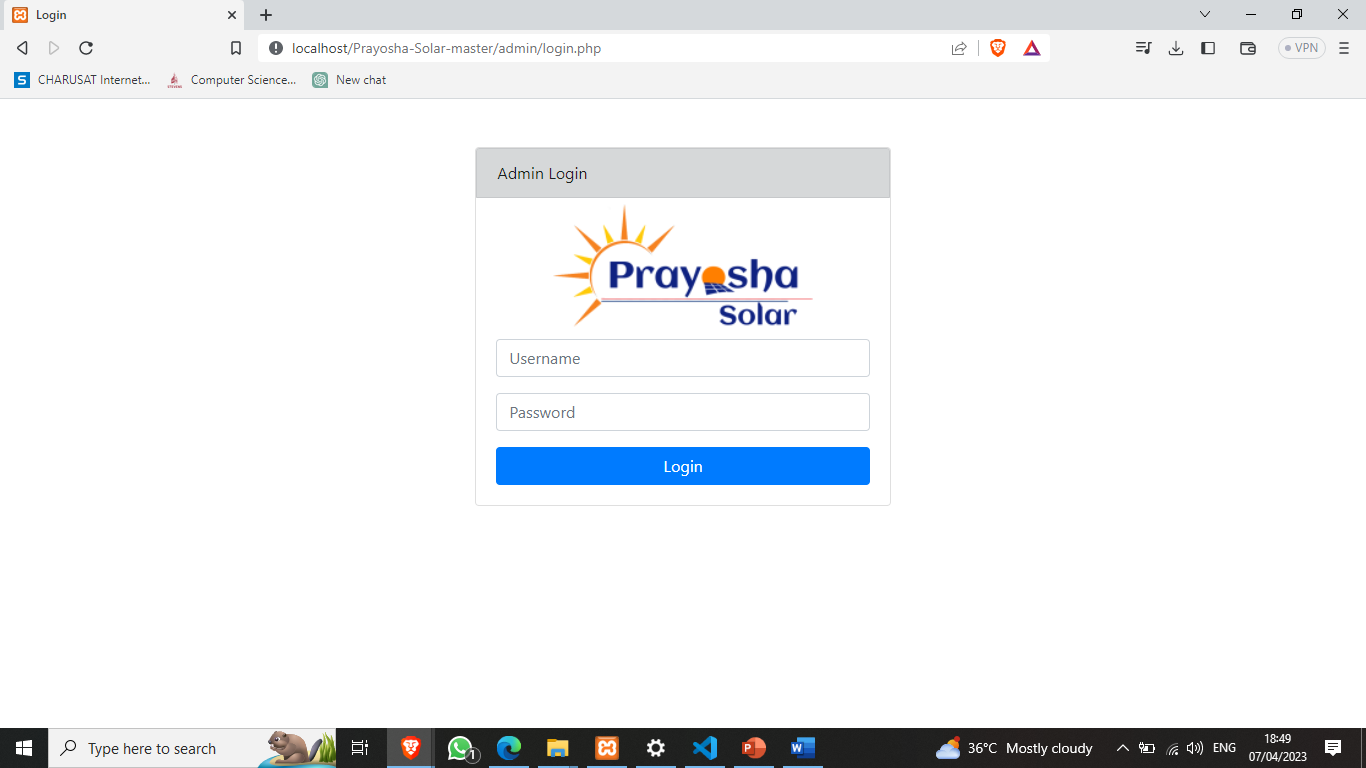
|  |  |
| --- | --- |
| Name: | Invoice Detail |
| Alias: | NULL |
| Where used?/ How used? | Generate Invoice on Purchase (Input/Output) |
| Content Description: | User id, product id, payment id, date |
| Supplementary Information: | User and Admin can see invoice.  Admin can generate invoice for the order. |

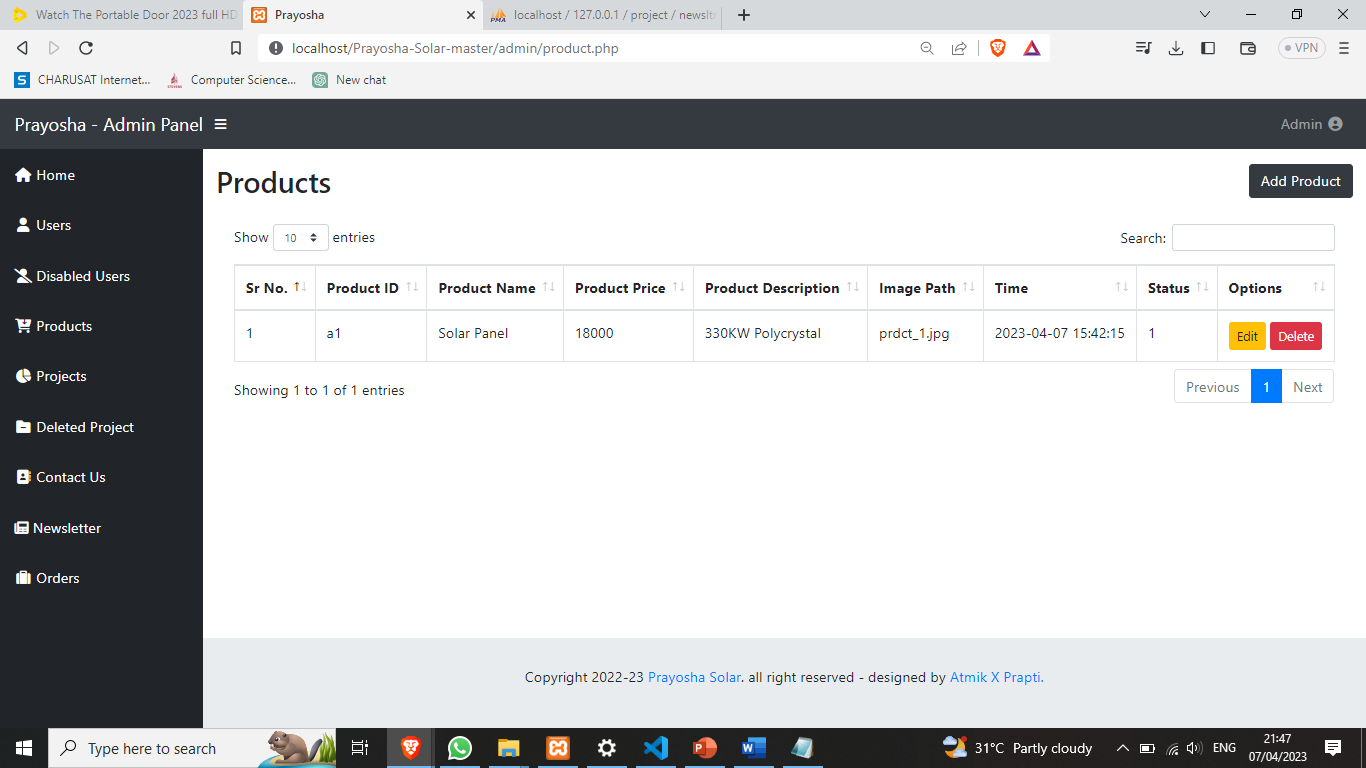
## Screen Layouts:

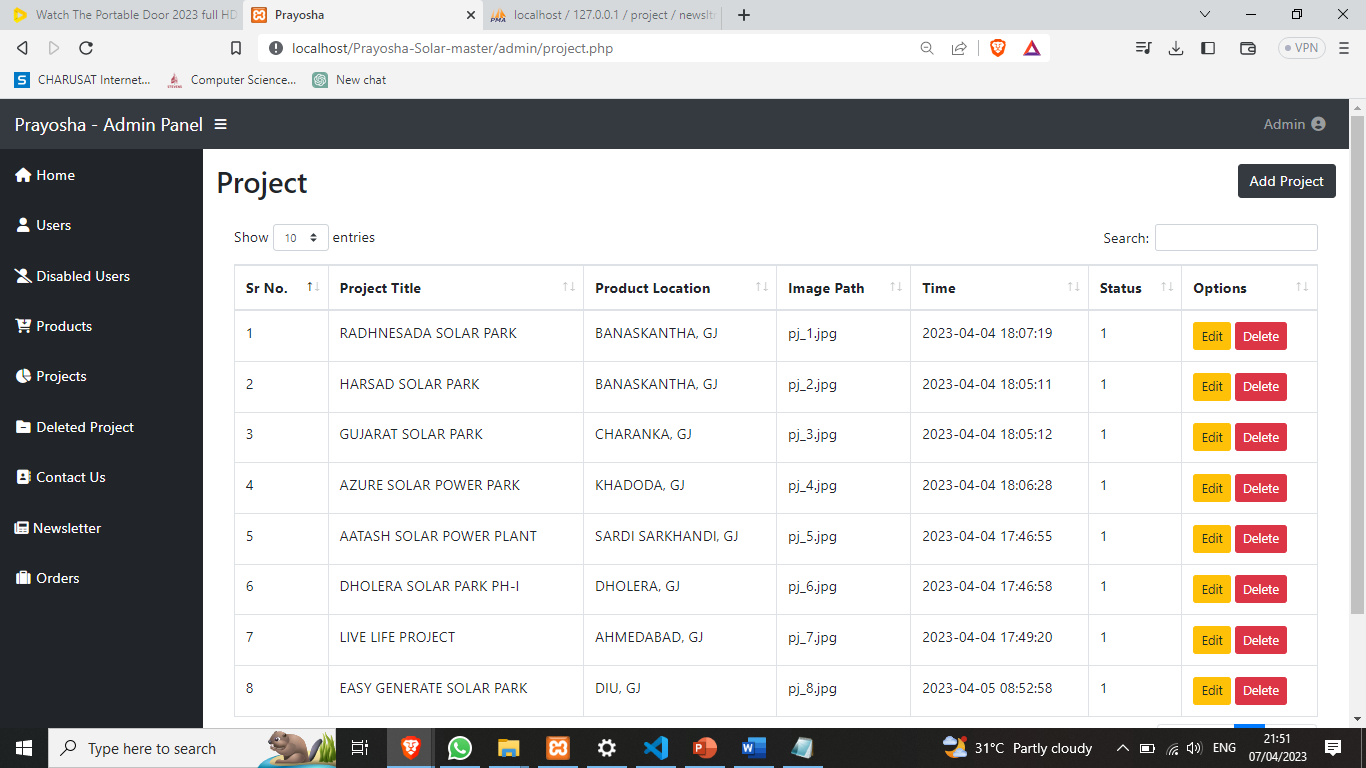
### Input Design

**[Admin Side Input Design]**

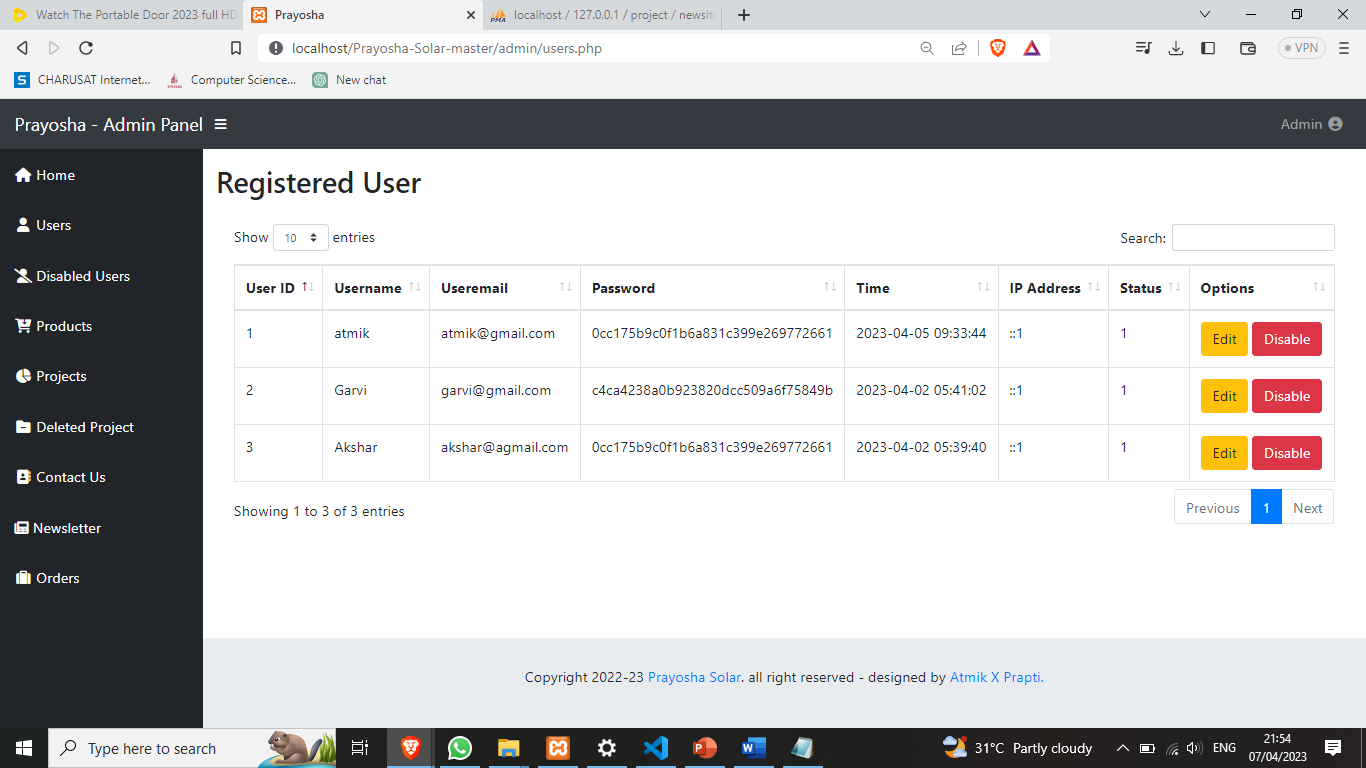
* Admin side Login



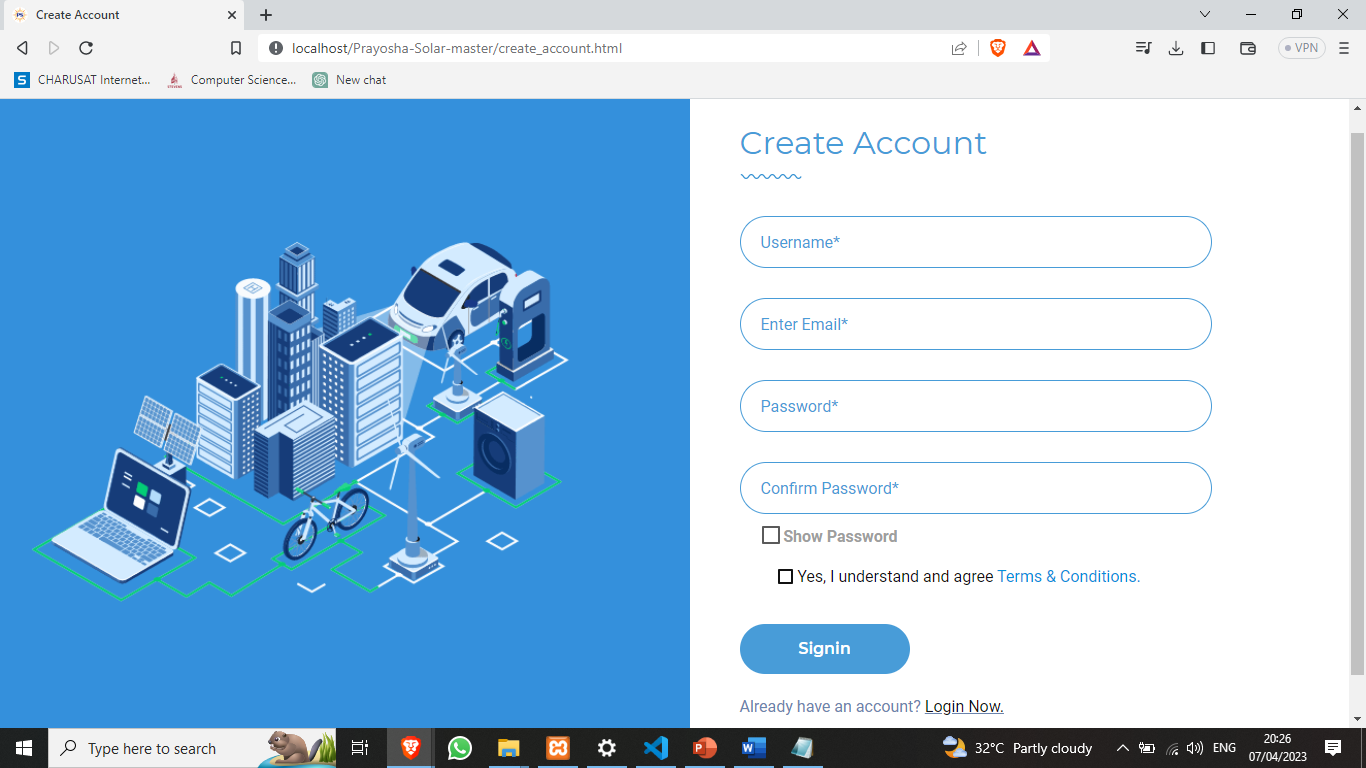
* Admin side product management
* Admin side add project

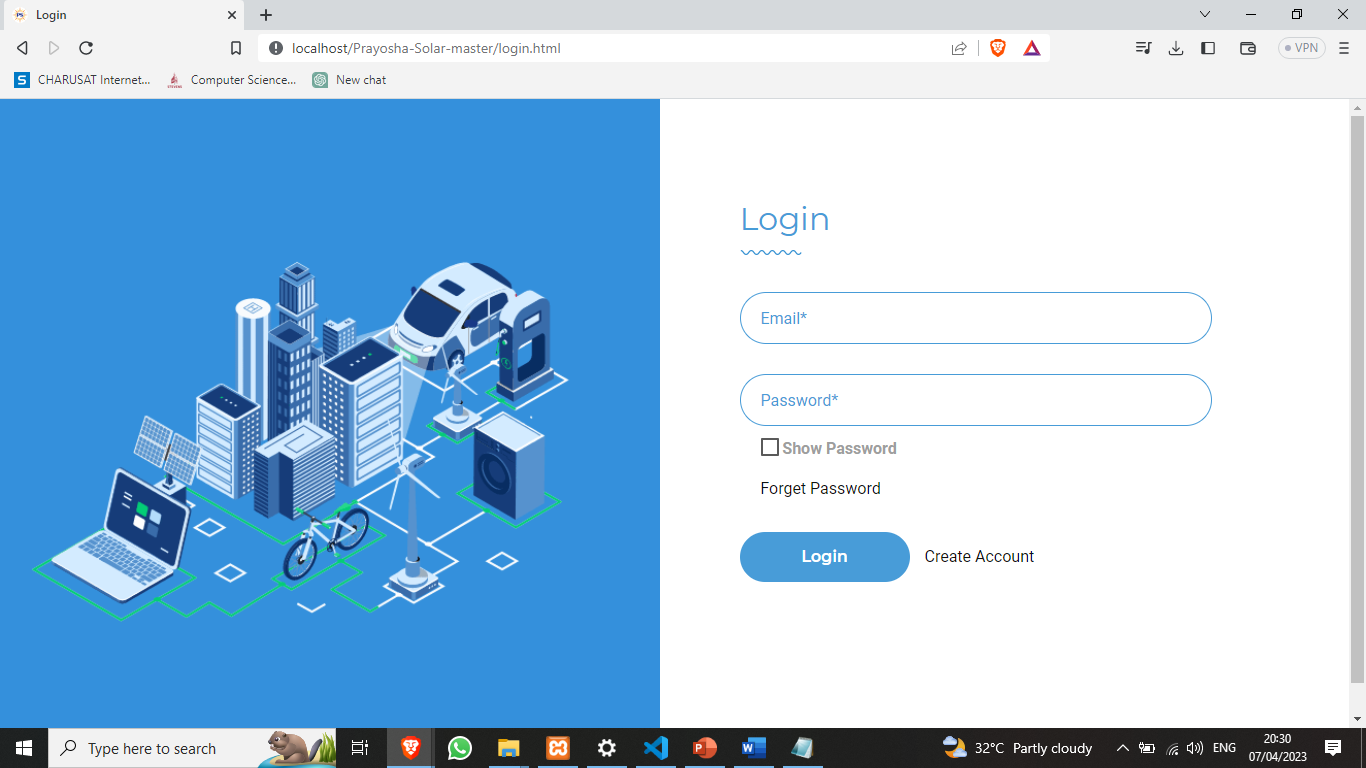


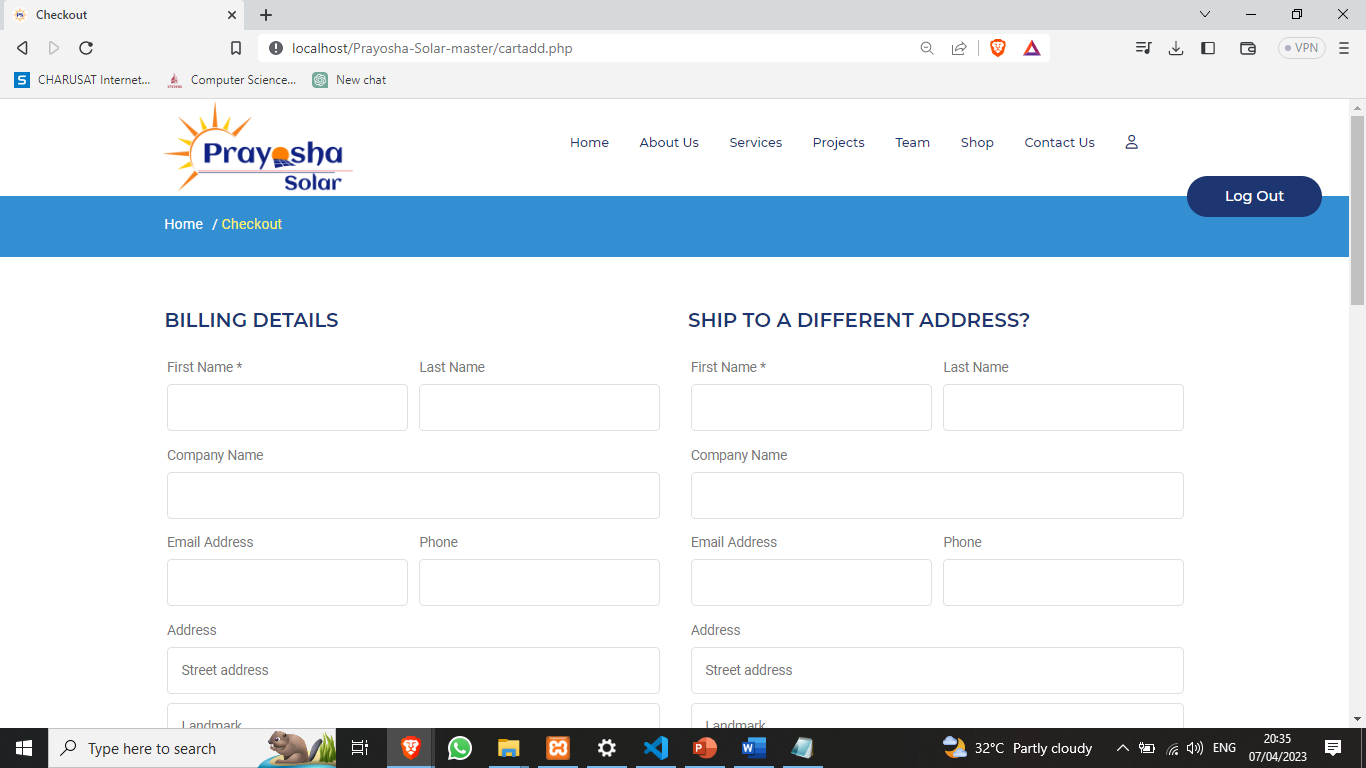
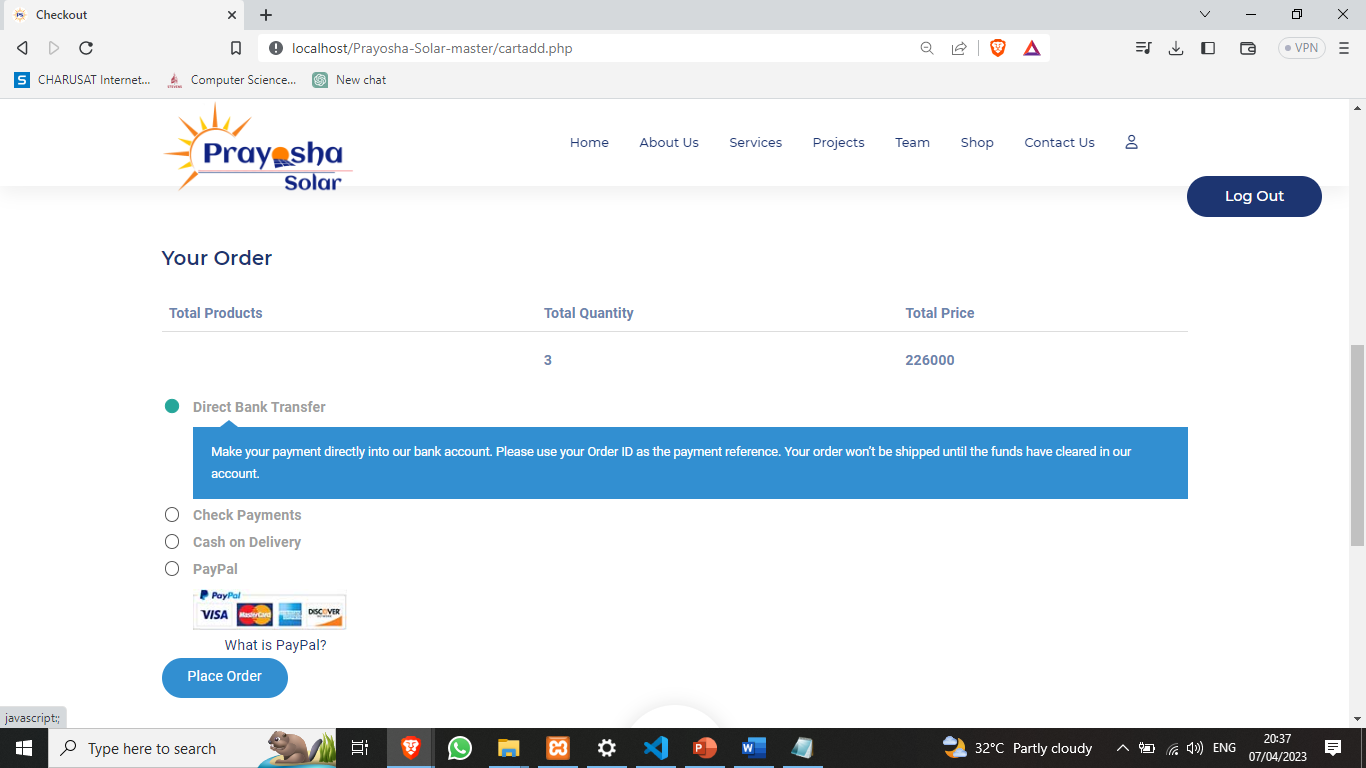
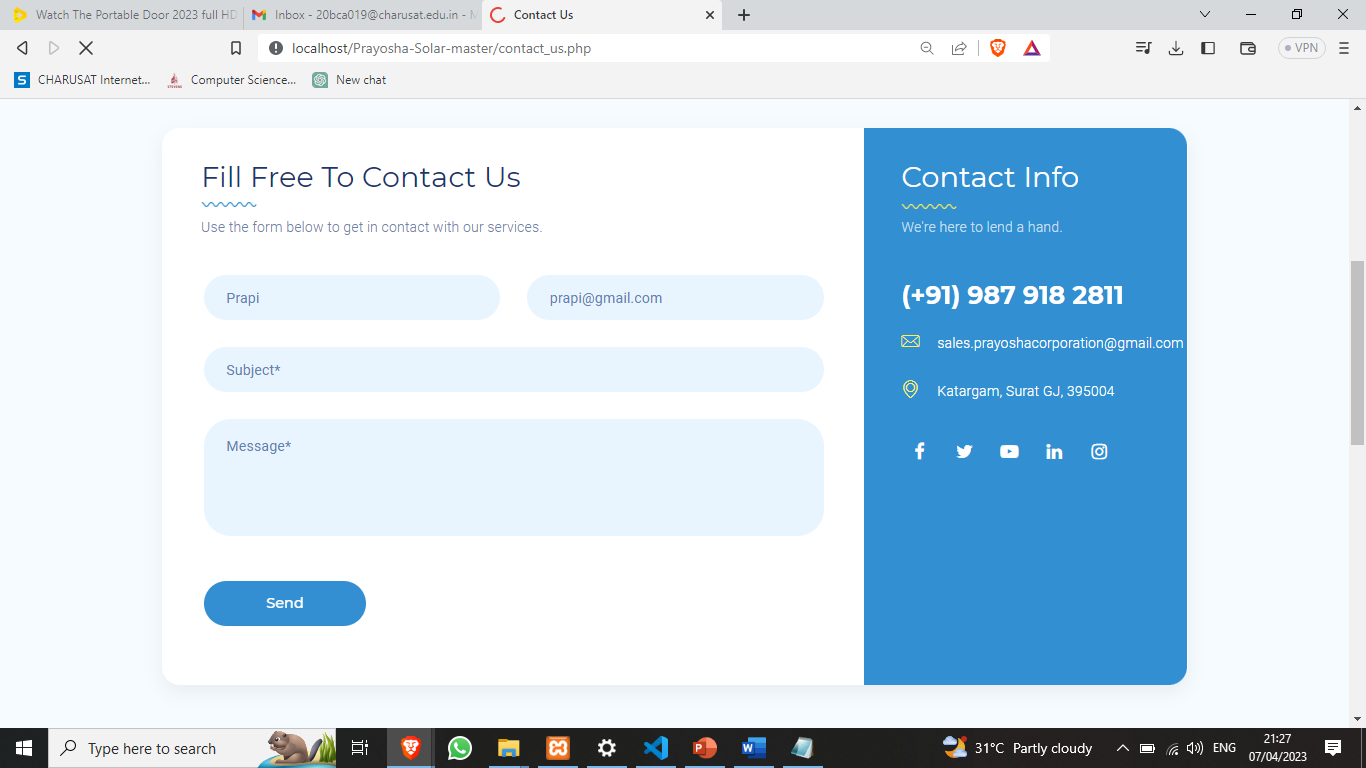
* Admin side edit user



**[User Side Input Design]**

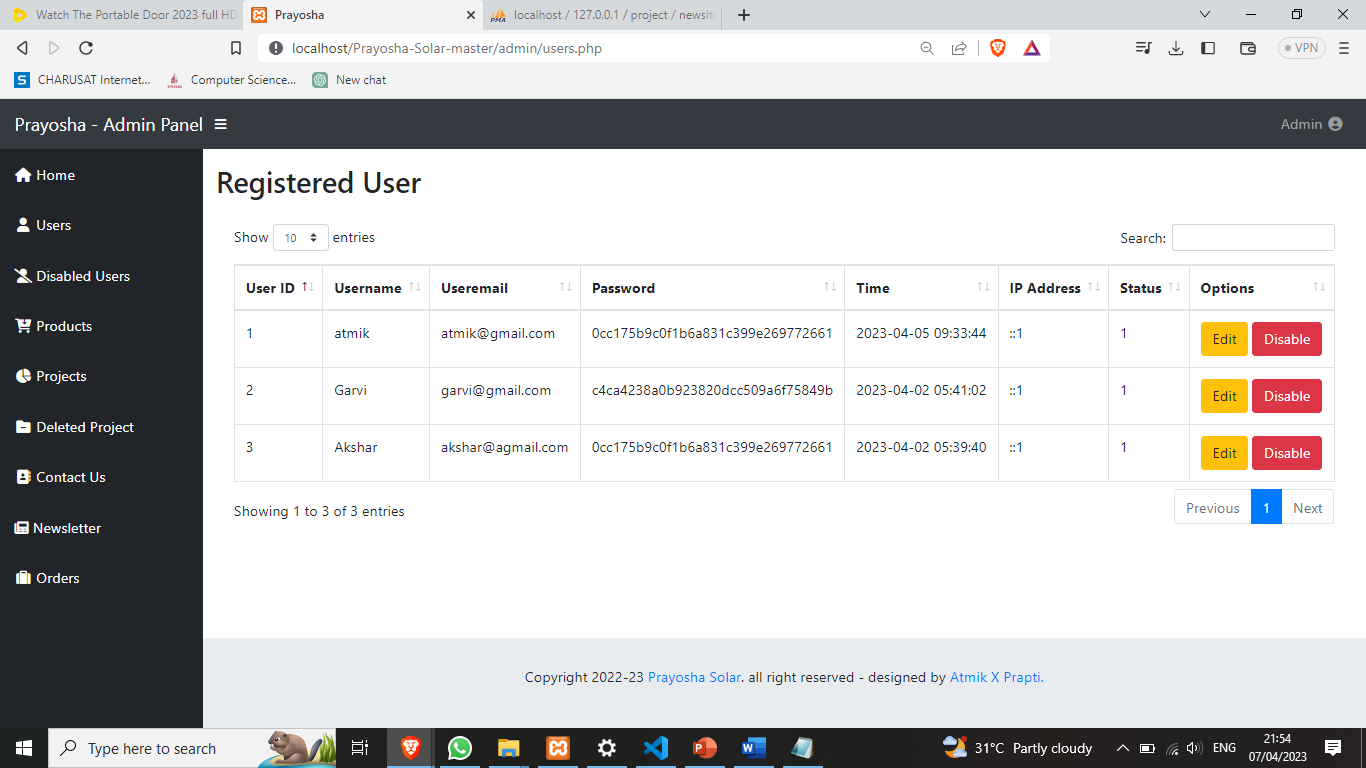
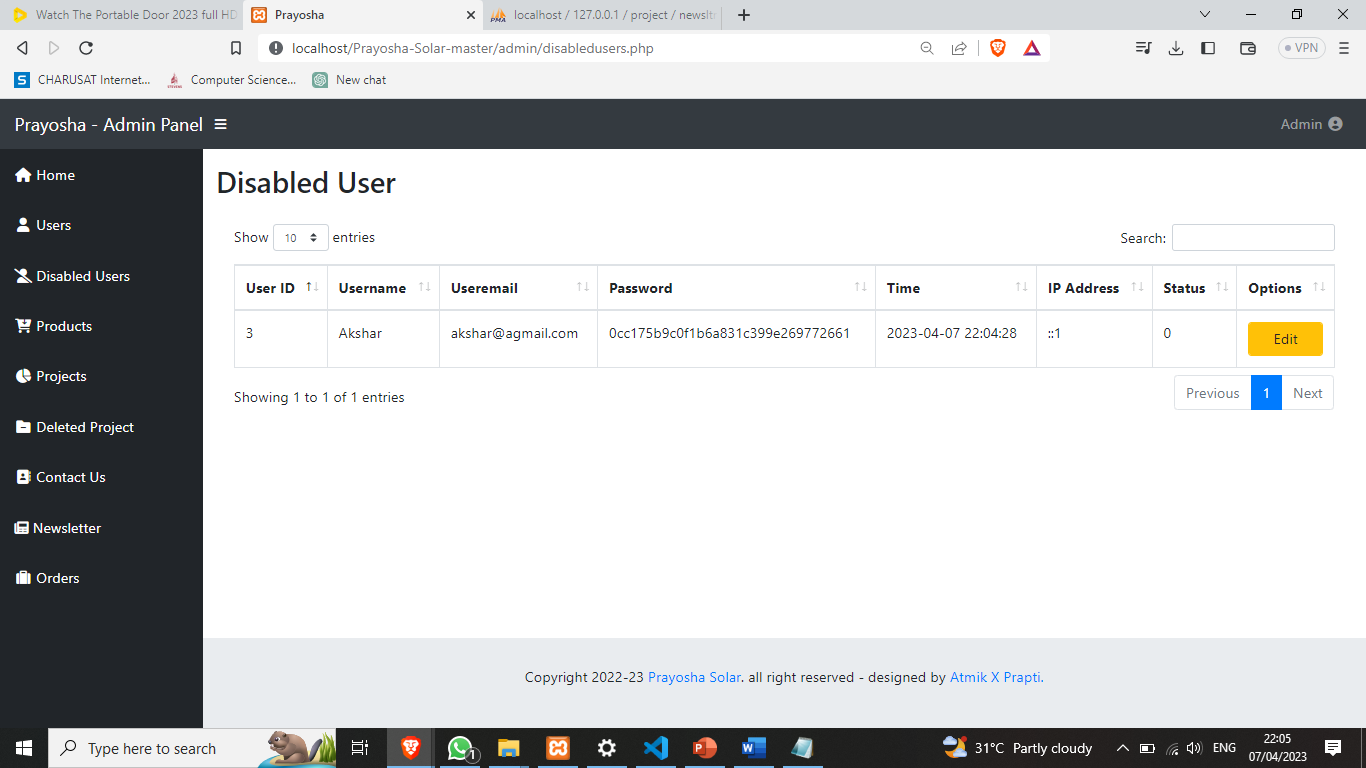
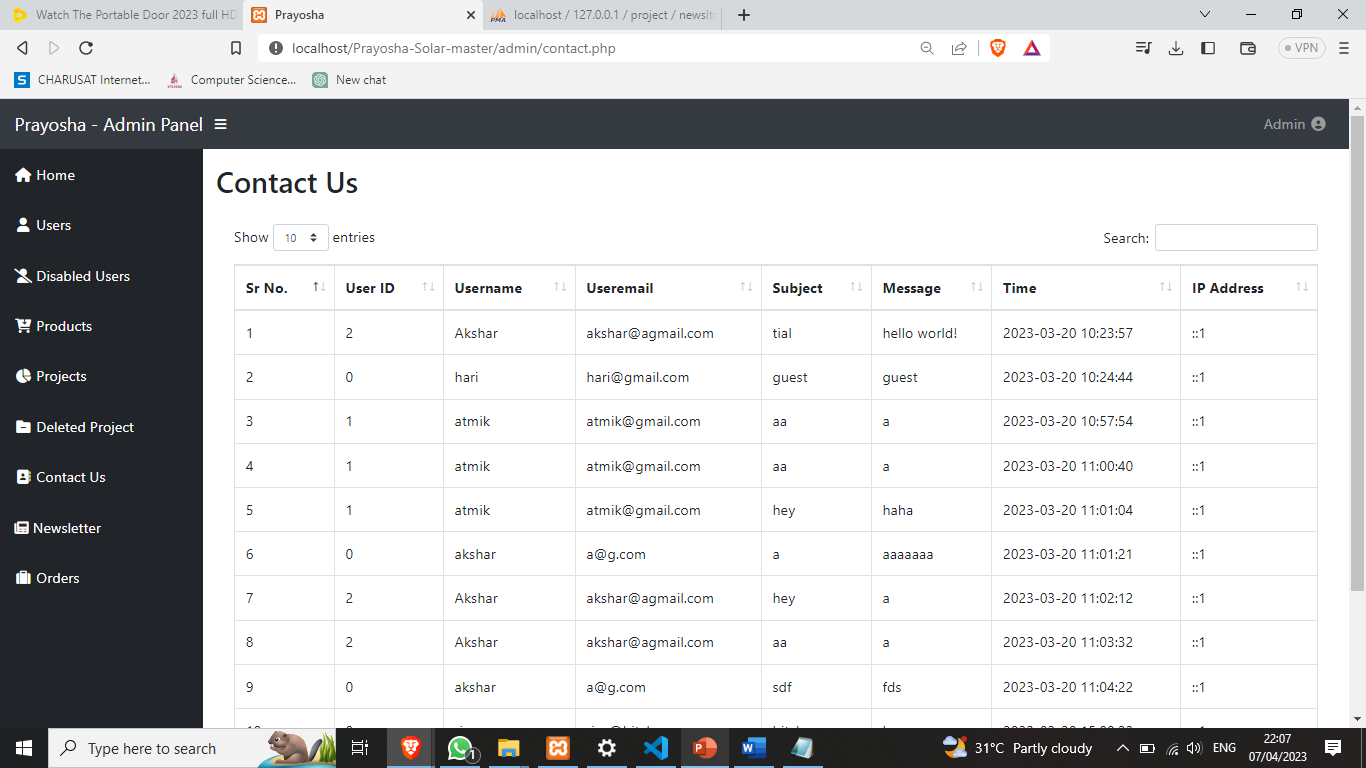
* User Side Registration
* User Side Login

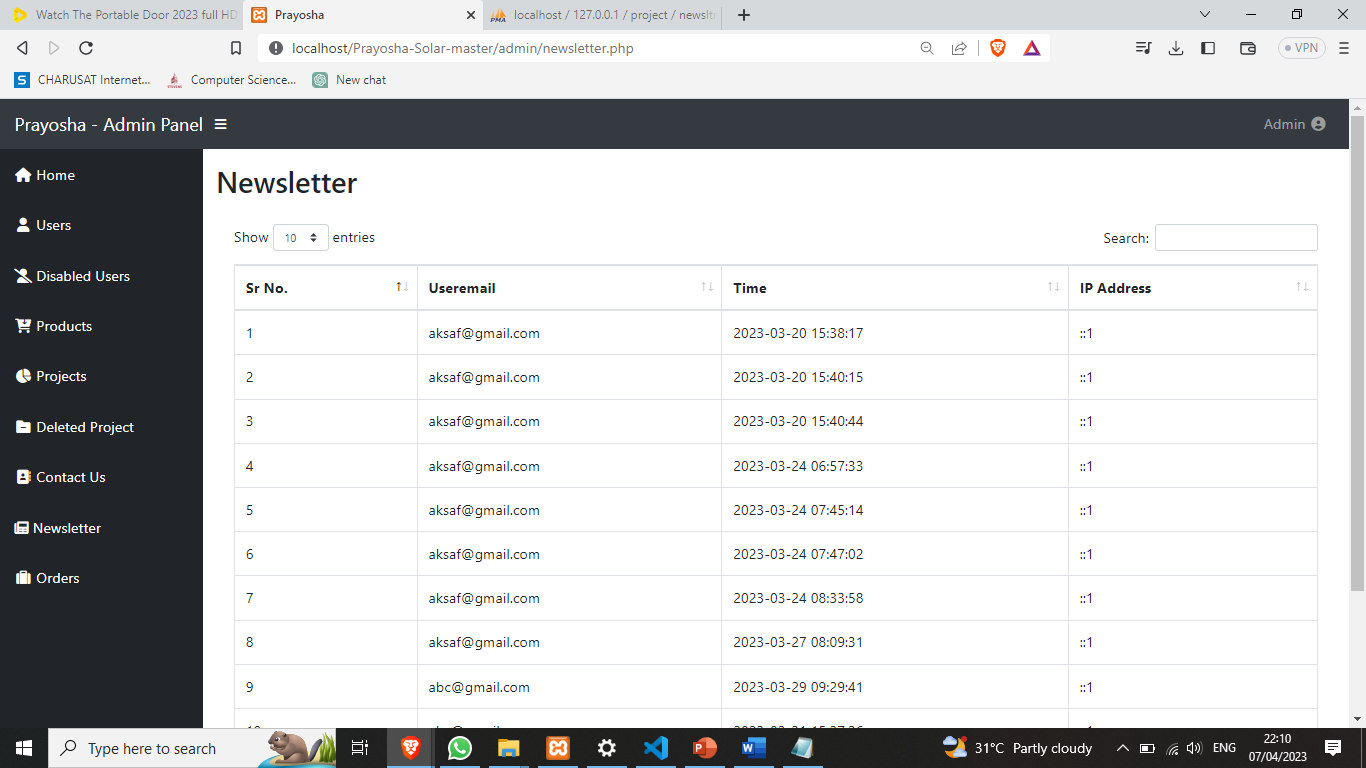


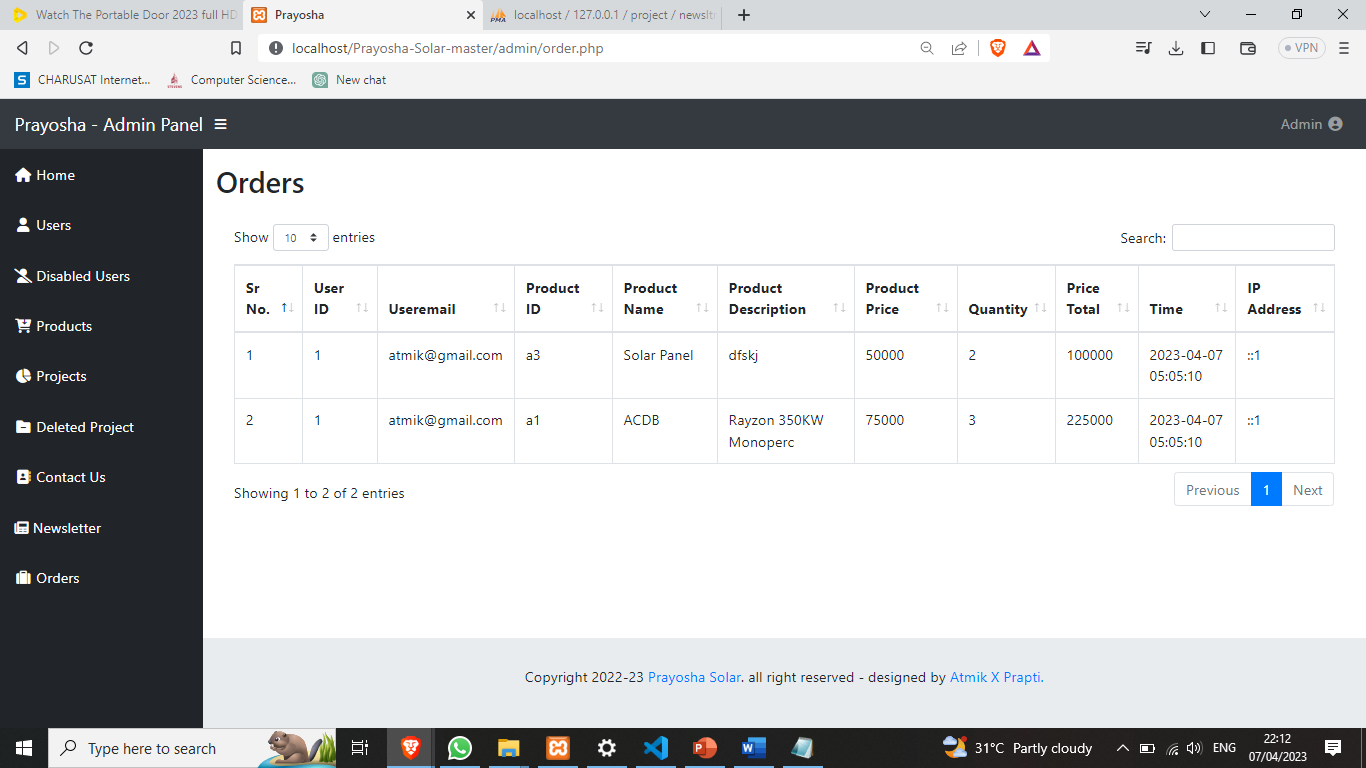
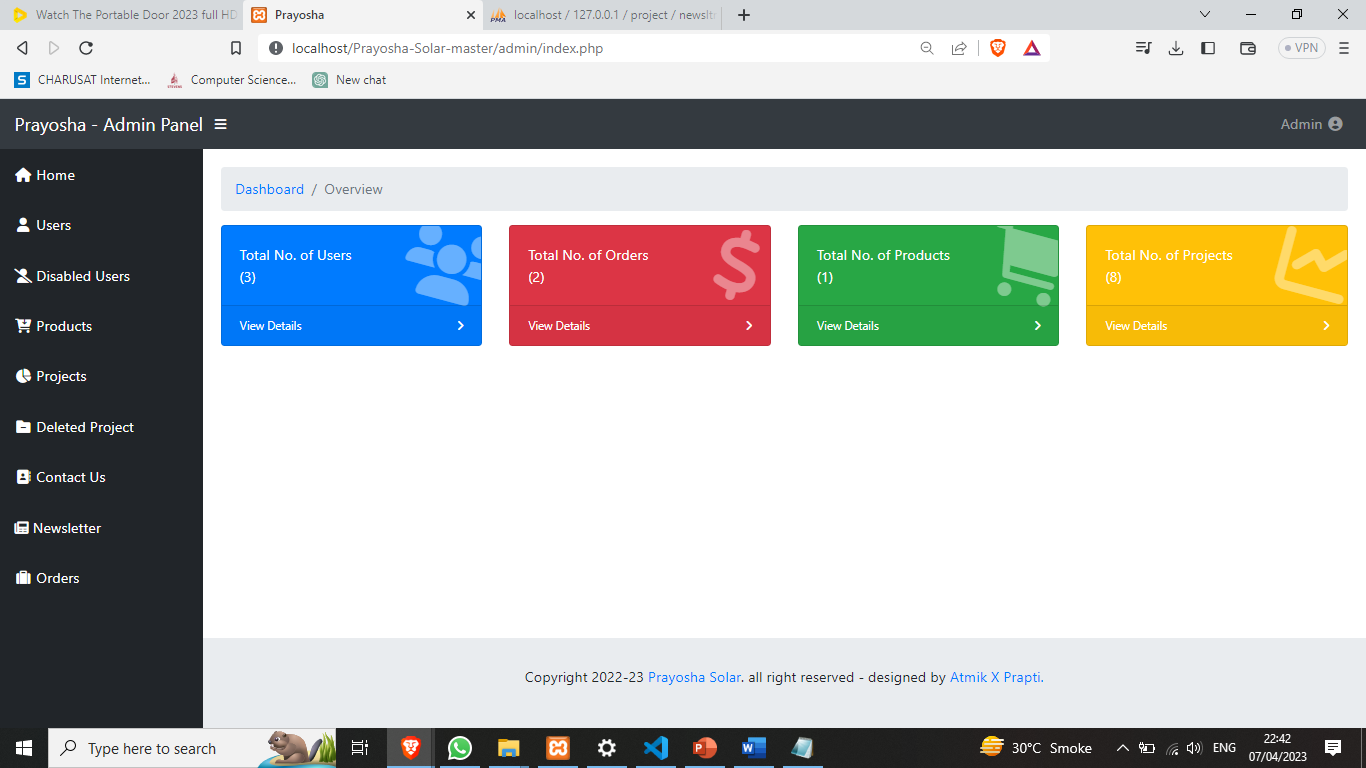
* User side Order Details
* User Side Payment Details
* User side contact panel

### Output Design

**[Admin Side Output Design]**

* Admin side user reports
* Admin side disabled users
* User side Contact details
* Admin side newsletter



* Admin Side order
* Admin side dashboard

# Software Testing

**[Admin]**

* **Login:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Field | Test Data | Valid / Invalid | Test Result |
| Email Id | Empty | Invalid | Please enter Email Id |
|  | shreya.com | Invalid | Please enter valid Email Id |
|  | shreya36@gmail.com | Valid | Inputted text is correct |
| Password | Empty | Invalid | Please enter password |
|  | shreya36 | Valid | Given password is same as exist in database |
| Login | Empty | Invalid | Please enter Email Id and password |
|  | [shreya@gmail.com](mailto:shreya@gmail.com)  shreya | Invalid | Email or password incorrect |
|  | [shreya36@gmail.com](mailto:shreya36@gmail.com)  shreya36 | Valid | If Email and password is correct than login will be successful |

* **Forgot Password:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Field | Test Data | Valid / Invalid | Test Result |
| Email Id | Empty | Invalid | Please enter Email Id |
|  | shreya.com | Invalid | Please enter valid Email Id |
|  | shreya36@gmail.com | Valid | Send OTP to Email |
| OTP | Empty | Invalid | Please enter password |
|  | 123456 | Invalid | Please enter valid OTP (if it does not match with database) |
|  | 134628 | Valid | Given OTP is same as that in database |
| New Password | Empty | Invalid | Please enter new password |
|  | shreya@369 | Valid | Inputted text is correct |
| Confirm Password | Empty | Invalid | Please enter new password again to confirm |
|  | shreya | Invalid | Password should the new password |
|  | shreya@369 | Valid | Password changed successfully |

* **Change Password:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Field | Test Data | Valid / Invalid | Test Result |
| Old Password | Empty | Invalid | Please enter old password |
|  | 12345 | Invalid | Please enter valid password (if the password does not match in database) |
|  | shreya@369 | Valid | Given password is same as in database |
| New Password | Empty | Invalid | Please enter new password |
|  | Shreya1357 | Valid | Inputted text is accepted |
| Confirm Password | Empty | Invalid | Please enter new password |
|  | 1357 | Invalid | Password mismatched (if password does not match new password) |
|  | Shreya1357 | Valid | Password changed successfully |

# Future Enhancement

* Future Enhancements:
* Info on current solar production
* Visualization and design of a virtual solar system
* Integrate energy storage
* Analysis and monitoring of energy use
* Social components and game mechanics
* Troubleshooting and remote supervision
* Individualized energy-saving advice
* Adaptation to smart house technology
* Improved customer service and chatbot help
* Report generation and analytics for solar installers.

# References and Bibliography

1. W3School (<https://www.w3school.com/>)
2. GitHub (<https://github.com/>)
3. Solar Energy International (SEI) (<https://www.solarenergy.org/>)
4. Solar Energy Corporation of India Limited (SECI) (<https://seci.co.in/>)
5. SQL Injection by PortSwigger (<https://portswigger.net/web-security/sql-injection>)
6. SQL Injection Defense by Acunetix (<https://www.acunetix.com/blog/sql-injection-defense/>)