**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **TOPIC NAME** | **PAGE NO** |
| **1** | **Introduction** | **2** |
| **2** | **Objective of Project** | **3** |
| **3** | **Purpose of Project** | **4** |
| **4** | **Software Design** | **5** |
| **5** | **Visibility Study** | **6** |
| **6** | **System Requirement** | **9** |
| **7** | **ER Diagrams** | **10** |
| **8** | **Use Case Diagram** | **11** |
| **9** | **Class Diagram** | **12** |
| **10** | **Screenshot of Code & Website** | **13** |
| **11** | **Advantages** | **24** |
| **12** | **Future Enhancement** | **25** |
| **13** | **Conclusion** | **26** |

**Introduction**

Technosky Solar System is a lightweight PHP-based web application developed for a solar equipment and installation company owned by Mr. Vishal Mane. The company provides solar panel installation, maintenance, and consultation services for both residential and commercial clients. To simplify day-to-day activities and reduce dependency on manual work, a centralized admin panel was planned to bring all operations under one system.

Before this system was developed, most tasks such as billing, material tracking, and client follow-ups were handled manually through Excel sheets, phone calls, and WhatsApp notes. This often resulted in missing data, delayed updates, difficulty verifying stock availability, and confusion about which materials were used for specific projects. Over time, these challenges started affecting workflow efficiency and record accuracy.

To overcome these limitations, a digital solution was created to automate the process. The system provides secure login access for the owner and one authorized employee. After logging in, they can easily add and manage client information, maintain product inventory along with pricing, track material usage, and monitor project progress. The system also supports bill generation with itemized line items, making it easier to create accurate and professional invoices.

All data is stored securely in a MySQL database, which helps keep records organized, reduces errors, and ensures that information is always available whenever needed. The application aims to make the company’s operations faster, more transparent, and easier to manage.

Key Modules

* Admin Authentication
* Client Management
* Product Inventory Management
* Project Management
* Billing with Line Items
* Reports & System Settings

**Objectives of the Project**

The main objective of the Technosky Solar System Admin Panel is to replace the manual process of managing clients, products, and billing with a simple and efficient digital system. The project is developed to help the company maintain accurate records and reduce time spent on repetitive tasks.

This system allows the admin to log in securely and perform all important operations from one place. Instead of handling data in separate books or Excel sheets, the owner and employee can now manage everything through a single dashboard. This improves accuracy and makes information easy to access whenever needed.

The specific objectives of the project are:

1. To provide an easy-to-use admin interface where the owner and staff can quickly access all records.
2. To maintain proper client records with contact details, project history, and billing information.
3. To manage product inventory and pricing, so that stock levels and material usage can be tracked.
4. To organize project details such as installation dates, assigned clients, and equipment used.
5. To generate bills with itemized line entries, helping in transparent and professional billing.
6. To store all data securely in a MySQL database for future reference and reporting.
7. To reduce manual errors that occur due to handwritten or scattered digital records.
8. To save time and improve work efficiency by automating routine tasks like billing and record management.

**Purpose of the Project**

The purpose of this project is to create a centralized and user-friendly system that helps Technosky Solar System manage its daily operations more efficiently. Instead of handling client records, product details, project information, and billing manually, this system provides a digital platform where everything can be stored, updated, and accessed in an organized way.

**The project aims to: -**

* Replace manual paperwork and Excel-based tracking with an automated solution.
* Ensure that all client, product, and billing information is stored in one secure place.
* Make billing faster and more professional by generating itemized invoices.
* Help the owner and staff save time by reducing repetitive tasks.
* Improve accuracy and prevent data loss, which is common in manual processes.
* Track which products are used in each project to maintain clear usage records and avoid confusion.

**Software Design**

**Overview**

This system is a simple PHP-based web application used to manage clients, products, projects, and billing for a solar company. Only admins can log in and access the dashboard. Once logged in, they can add clients, update inventory, assign projects, and generate itemized bills. The focus of the design is simplicity, easy maintenance, and reliability.

**Architecture**

* The system is monolithic, meaning everything runs in a single application.
* Pages are built using Core PHP with server-side rendering.
* HTML, CSS, and minimal JavaScript are used for the frontend.
* Data is stored in a MySQL database, accessed using PDO for security.
* The file structure is kept simple with folders like public/, includes/, css/, and sql/.

**Data Model (Basic Tables)**

* Admins: id, name, email, password
* Clients: id, name, email, phone, address
* Products: id, name, price, stock
* Projects: id, name, client\_id, status
* Bills: id, bill\_no, client\_id, project\_id, date, total
* Bill\_Items: id, bill\_id, product\_id, qty, price, line\_total

**Key Design Decisions**

* Use of PDO: Prevents SQL injection and supports different databases.
* Simple file-based structure: Easy to understand and modify.
* Foreign keys in database: Ensures data consistency.
* Minimal JavaScript: Only used for features like form validation and password toggle.

**Security & Integrity**

* Passwords are stored using password hashing (password\_hash).
* Prepared statements are used for all database queries.
* Admin access is controlled through server-side session checks.
* Regular database backups are taken to prevent data loss and ensure records can be restored if needed.

**Feasibility Study**

Before developing the Technosky Solar System Admin Panel, a basic feasibility study was conducted to check whether this project is practical and beneficial. The study covers four main aspects:

**1. Technical Feasibility**

* The application is built using PHP and MySQL, which are easily available and supported on most hosting platforms.
* No advanced hardware or expensive tools are required.
* Any basic computer with XAMPP / LAMP can run the system smoothly.
* Therefore, the project is technically feasible.

**2. Economic Feasibility**

* Development cost is low as the system is built using open-source technologies.
* It reduces manual work, saves time, and prevents mistakes in billing and record-keeping.
* The cost of development is easily justified by long-term savings and operational efficiency.
* Hence, the project is economically viable.

**3. Operational Feasibility**

* The system is simple enough for non-technical staff to operate.
* Only two users (Owner and Employee) will manage data, which makes training easy.
* Since it replaces scattered Excel sheets and manual billing, users will prefer the new system.
* Therefore, it is operationally feasible.

**4. Time Feasibility**

* The scope of the project is limited to basic modules like clients, products, projects, and billing.
* It can be developed and deployed in a short duration without delaying business operations.
* As a result, the project is time-feasible.

**System Requirements**

The Technosky Solar System Admin Panel is designed to run on a standard desktop or laptop with moderate specifications. The system requires a computer with at least a dual-core processor and 2 GB of RAM, though 4 GB or more is recommended for smoother performance. A minimum of 500 MB of free storage space is needed to install the application and store its data, while higher storage is preferable for future records. The display should support a resolution of 1024×768 pixels or above to ensure that the interface elements are visible and easy to interact with.

**1. Hardware Requirements**

• Dual-core processor or higher  
• Minimum 2 GB RAM (4 GB recommended)  
• At least 500 MB free storage  
• Display resolution 1024×768 or above

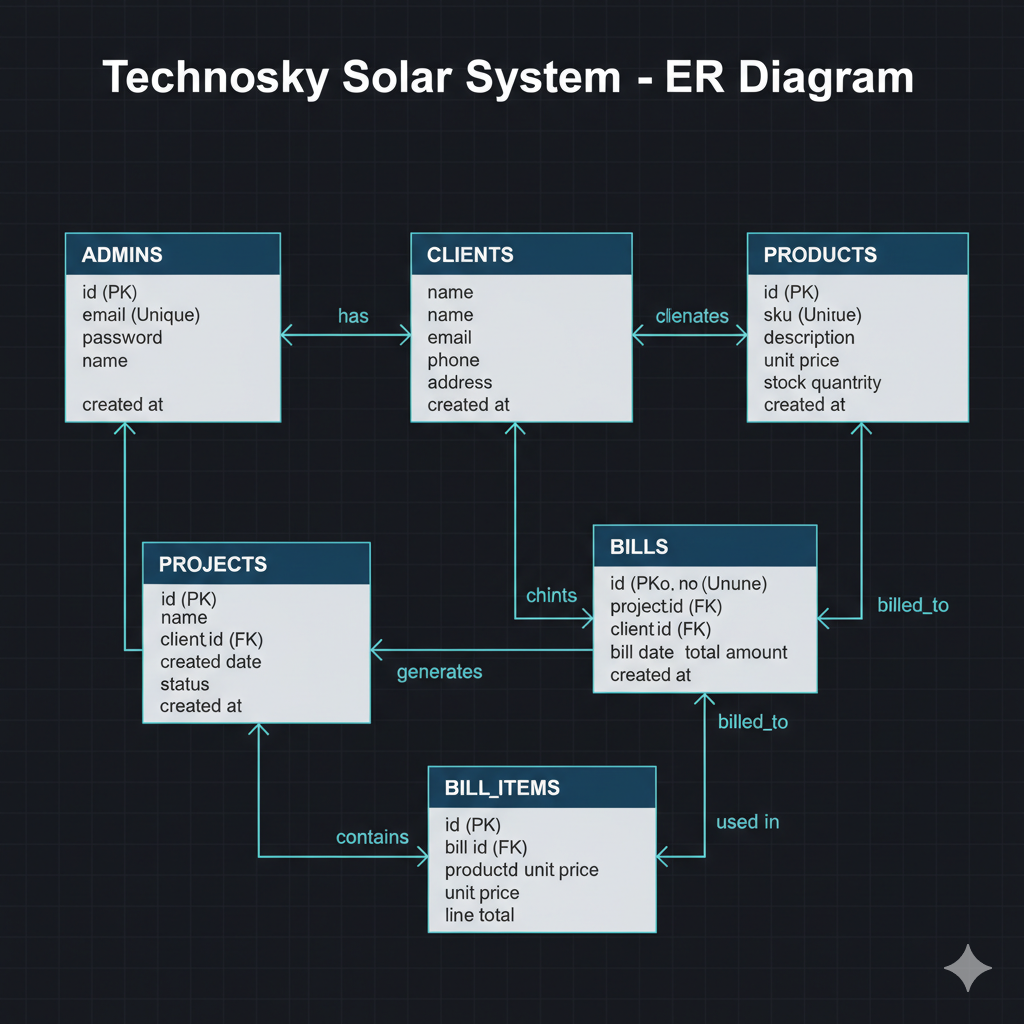
On the software side, the system runs on a web server environment like XAMPP, WAMP, or LAMP, which includes Apache and PHP 7.4 or higher. The database is handled using MySQL, with PDO support enabled for secure and flexible database operations. The frontend of the system is accessible through modern web browsers such as Chrome, Firefox, or Edge, which are recommended to ensure proper rendering of pages and forms.

**. Software Requirements**

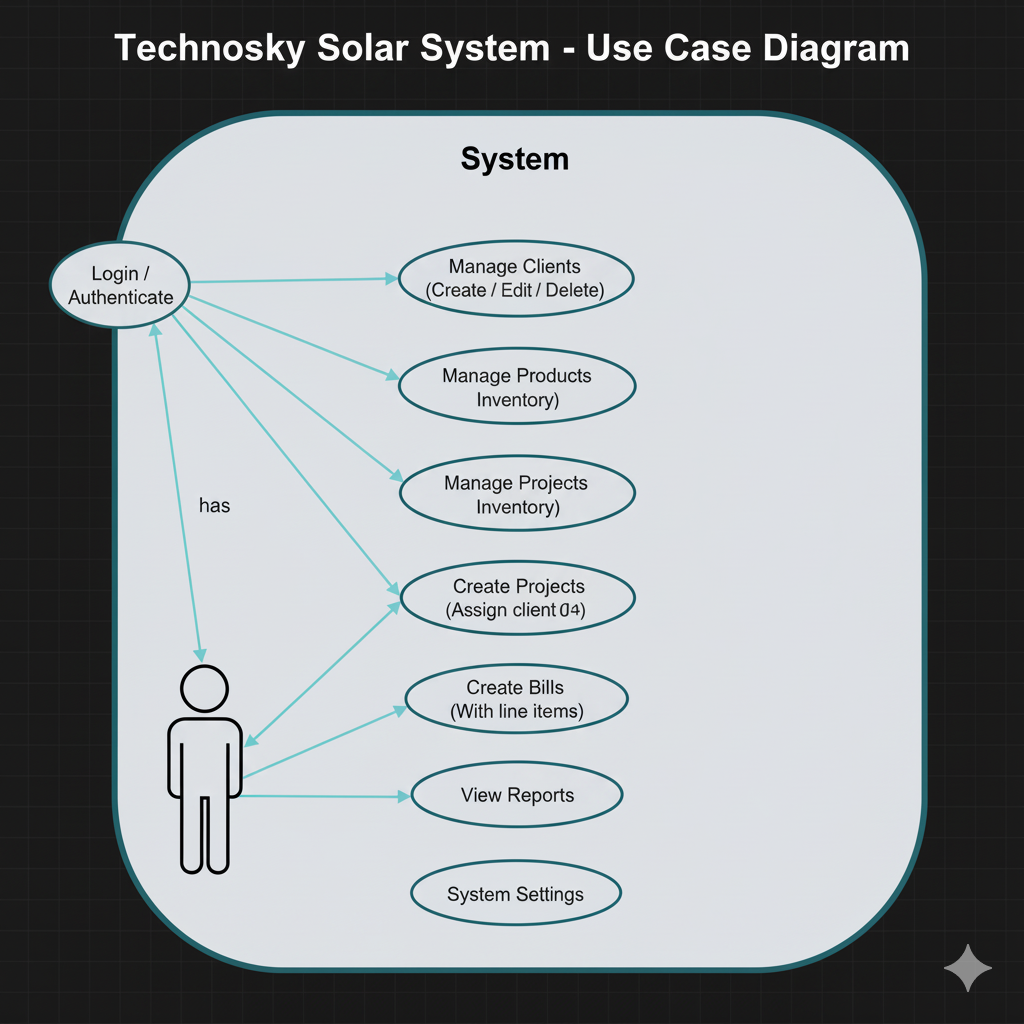
• XAMPP / WAMP / LAMP server  
• Apache server  
• PHP 7.4 or higher  
• MySQL with PDO enabled  
• Modern browser: Chrome, Firefox, or Edge

From the user perspective, only basic computer knowledge is required to operate the system. Users need access to the system through their local network or internet if the application is hosted online. Admin login credentials are necessary to access the dashboard and perform any operations, ensuring security and controlled access. Overall, the system is lightweight and designed to work efficiently on commonly available hardware and software, making it easy to install, use, and maintain for daily business operations.

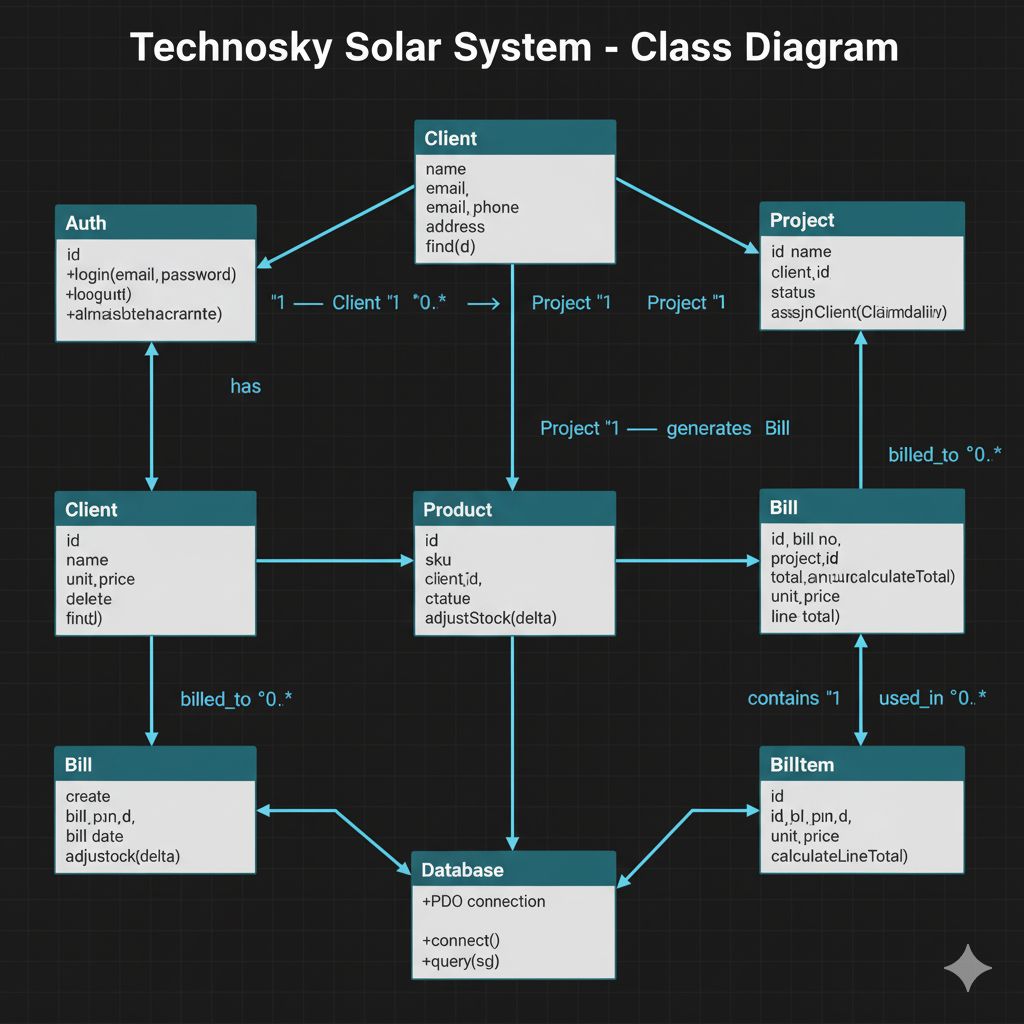
**ER Diagram**

****

**Use Case Diagram**

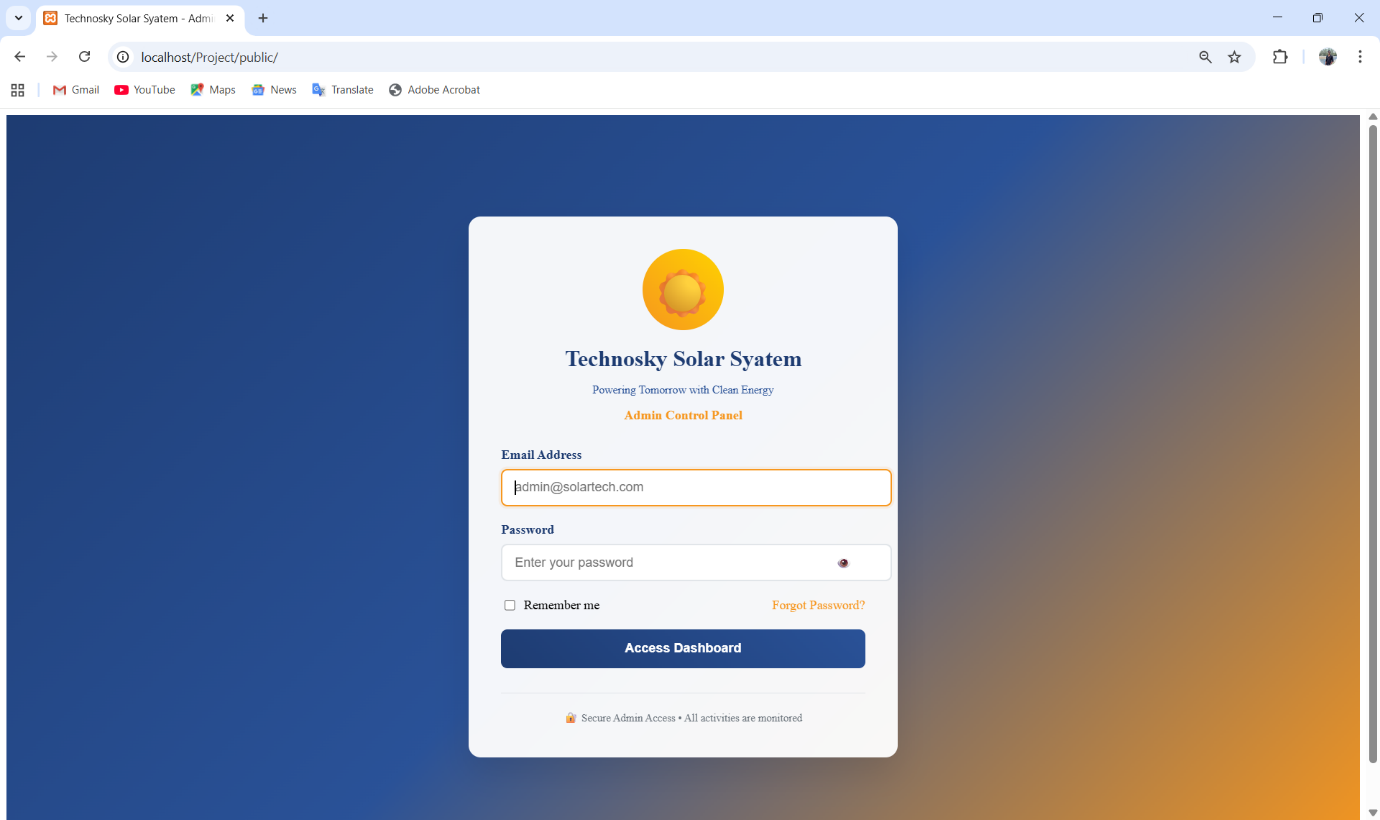
****

**Class Diagram**

****

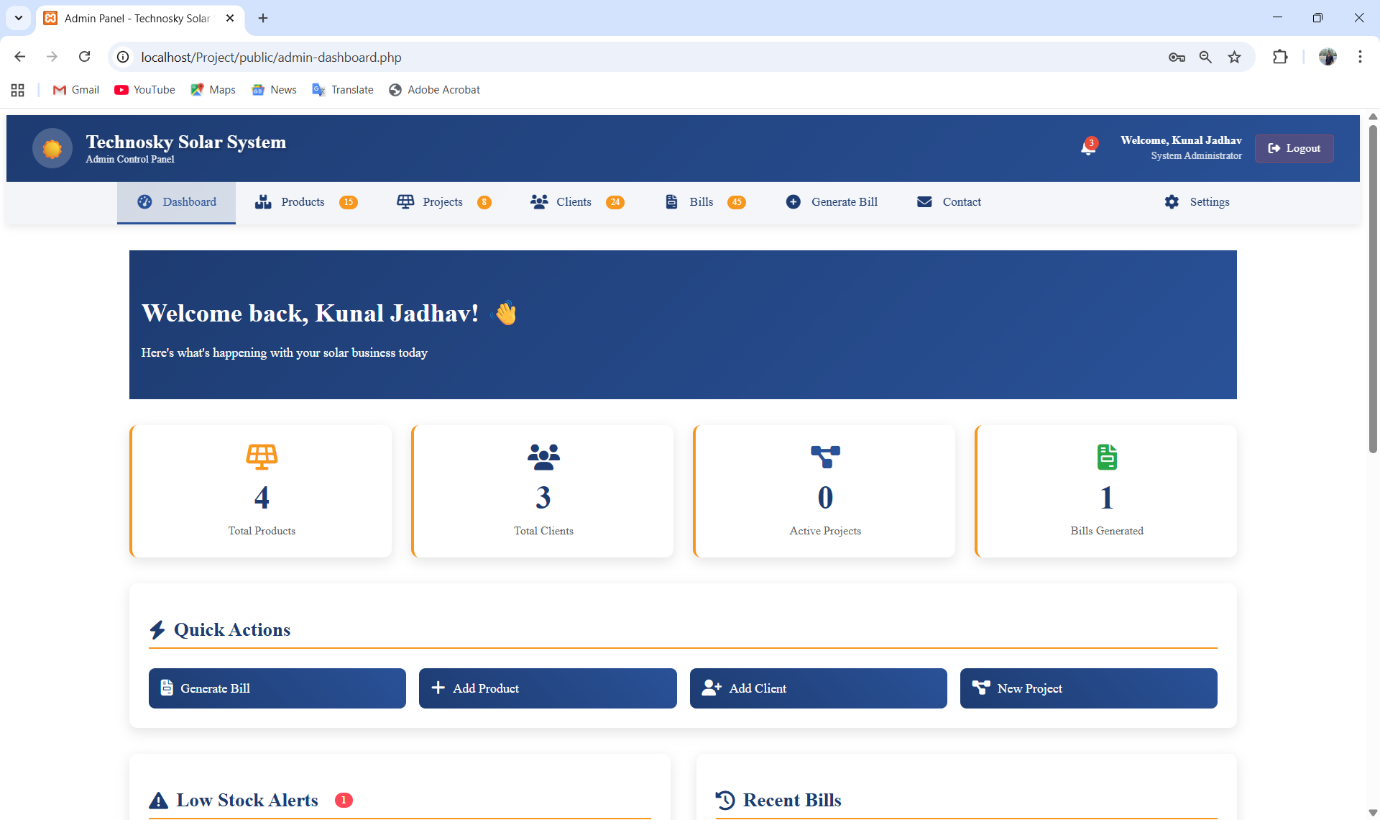
**Screenshot of Website**

**Login page**

****

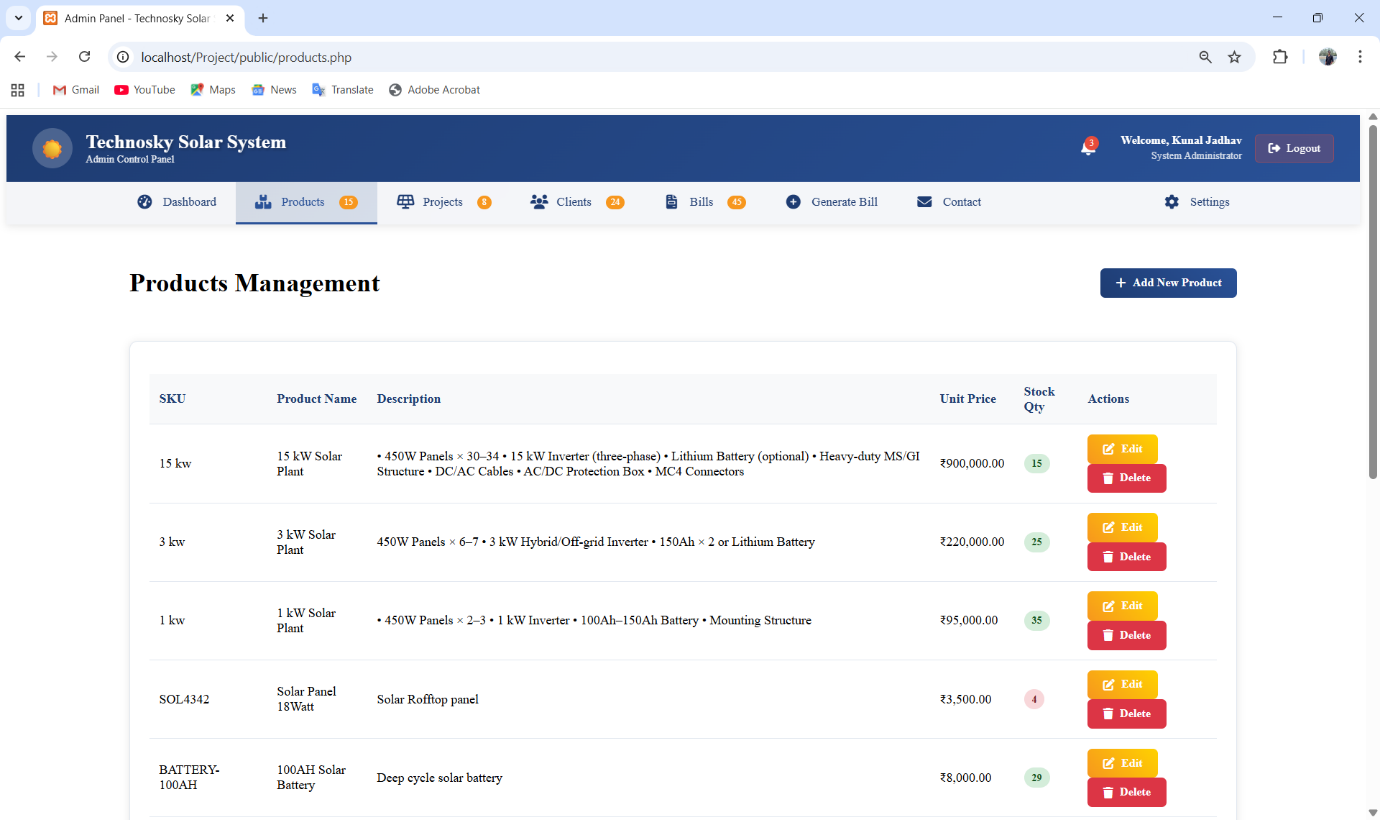
The Login Page is designed with a modern, gradient background transitioning from deep blue to a warm orange, reflecting the "Solar" theme of the company. At the center sits a clean, white card housing the authentication form. The branding is prominent at the top, featuring a sun icon and the company name "Technosky Solar System" followed by the tagline "Powering Tomorrow with Clean Energy." This reinforces brand identity before the user even accesses the system. The form specifically requests an Email Address and Password, with the email field currently pre-filled with an admin address.

**Dashboard**

****

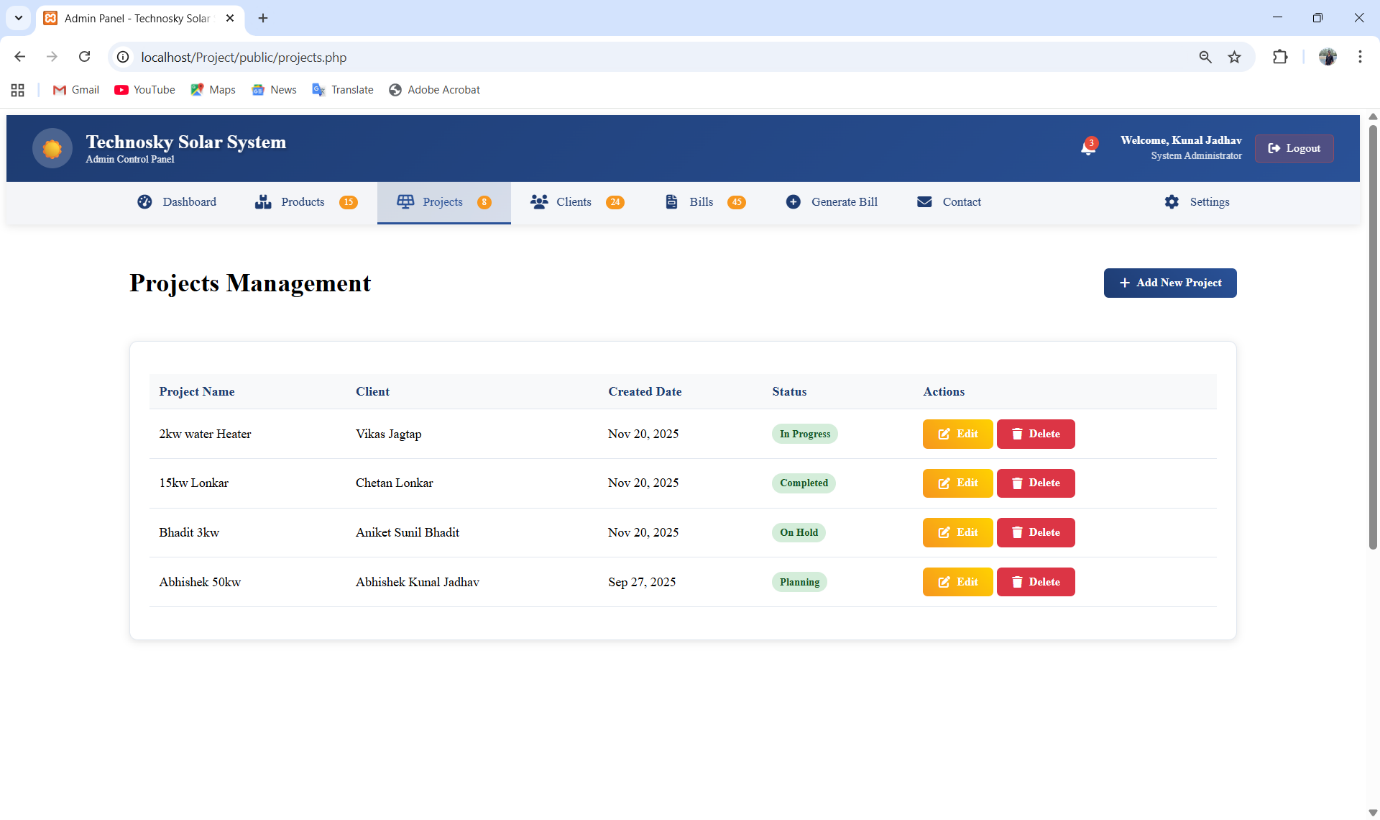
The Dashboard serves as the central command centre for the application, offering an immediate, high-level overview of the business's performance. The page features a clean, professional header with a dark blue navigation bar containing links to key modules like Products, Projects, Clients, and Bills, alongside a notification bell and a user profile for "Kunal Jadhav." A prominent "Hero" section welcomes the user back with a friendly greeting and a summary tagline. Below this, four distinct "Stat Cards" provide real-time quantitative data: Total Products (4), Total Clients (3), Active Projects (0), and Bills Generated (1); these cards use simple iconography to make data scannable at a glance.

**Product**



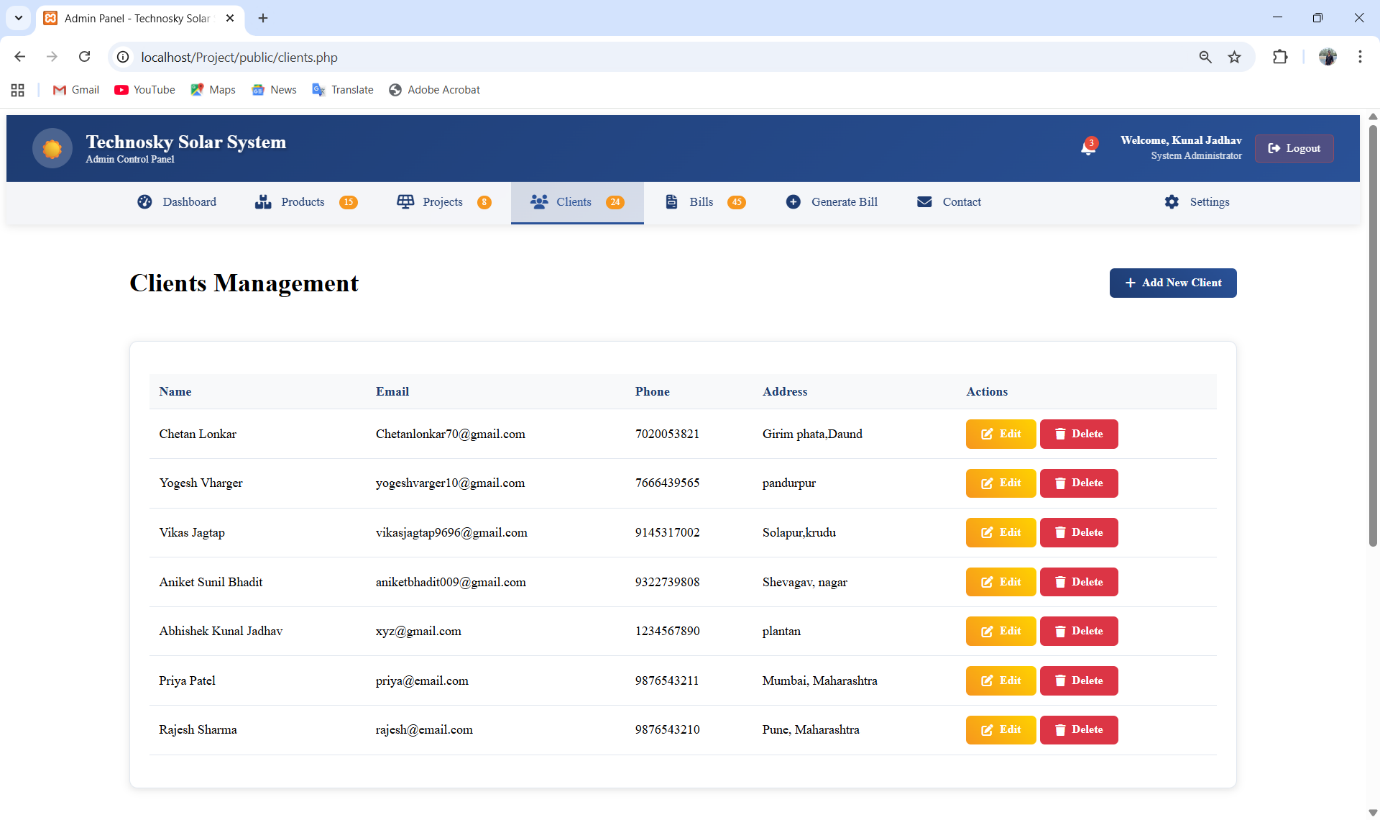
The **Products Management** page serves as the inventory ledger for the business. It provides a detailed view of the hardware available for installation, listing items by **SKU**, **Product Name**, **Description**, **Unit Price**, and **Stock Qty**. The descriptions are notably detailed, including technical specifications like "450W Panels," "Lithium Battery," and "Three-phase Inverters," which helps the admin distinguish between similar stock items. The pricing column is formatted clearly with currency symbols, and the stock quantity is highlighted in green bubbles (e.g., "15", "25"), providing a quick visual check of inventory levels.

**Projects**

****

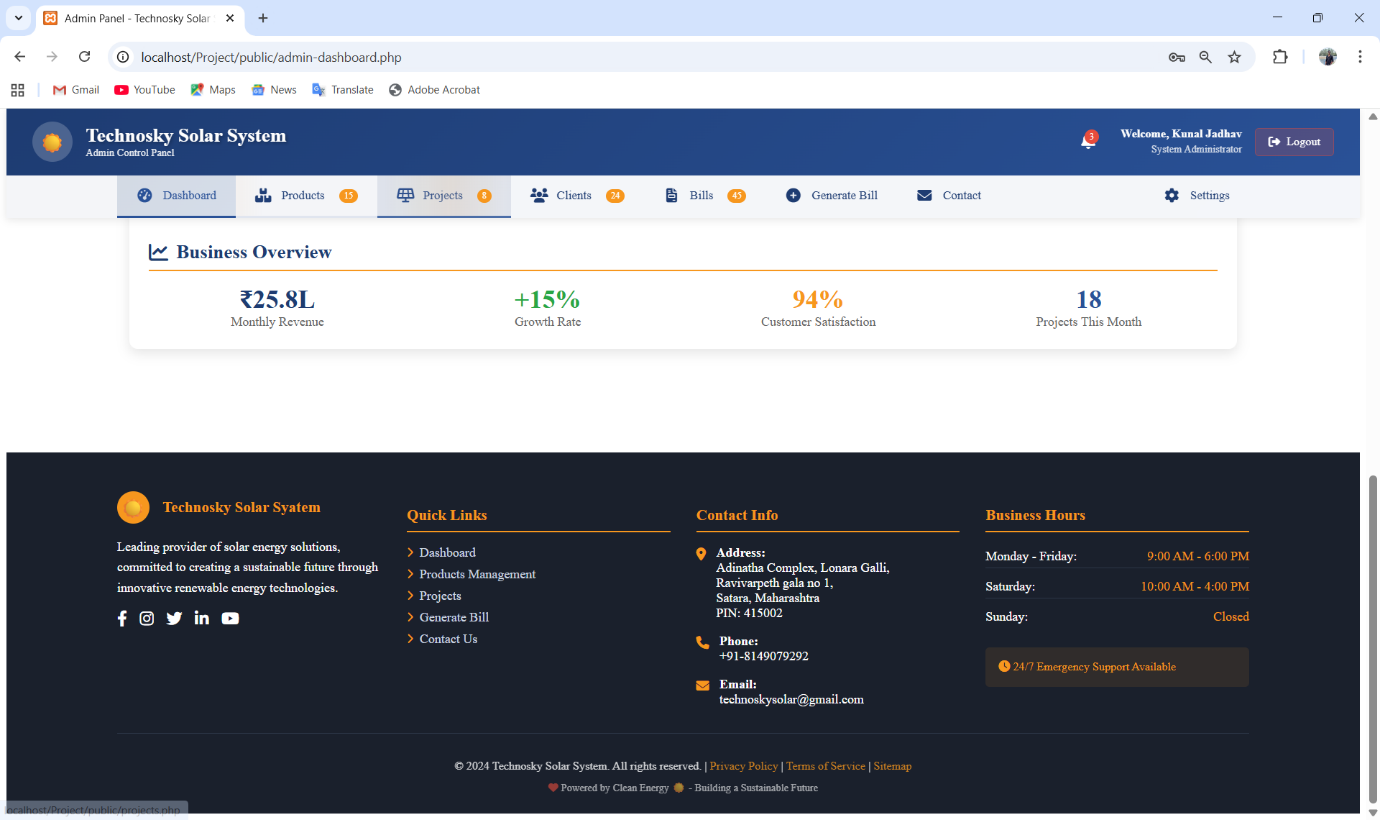
The Projects Management page acts as a tracking board for all solar installation jobs. It features a data table that lists essential details: Project Name, Client Name, Created Date, Status, and Actions. The table currently displays four diverse projects, ranging from a "2kw water Heater" to a larger "Abhishek 50kw" system, showing the system's capability tohandle various project scales. A key feature of this page is the color-coded Status badges—"In Progress" (green), "Completed" (teal), "On Hold" (greenish-grey), and "Planning" (grey)—which allow the admin to assess workflow bottlenecks instantly.

**Client**

****

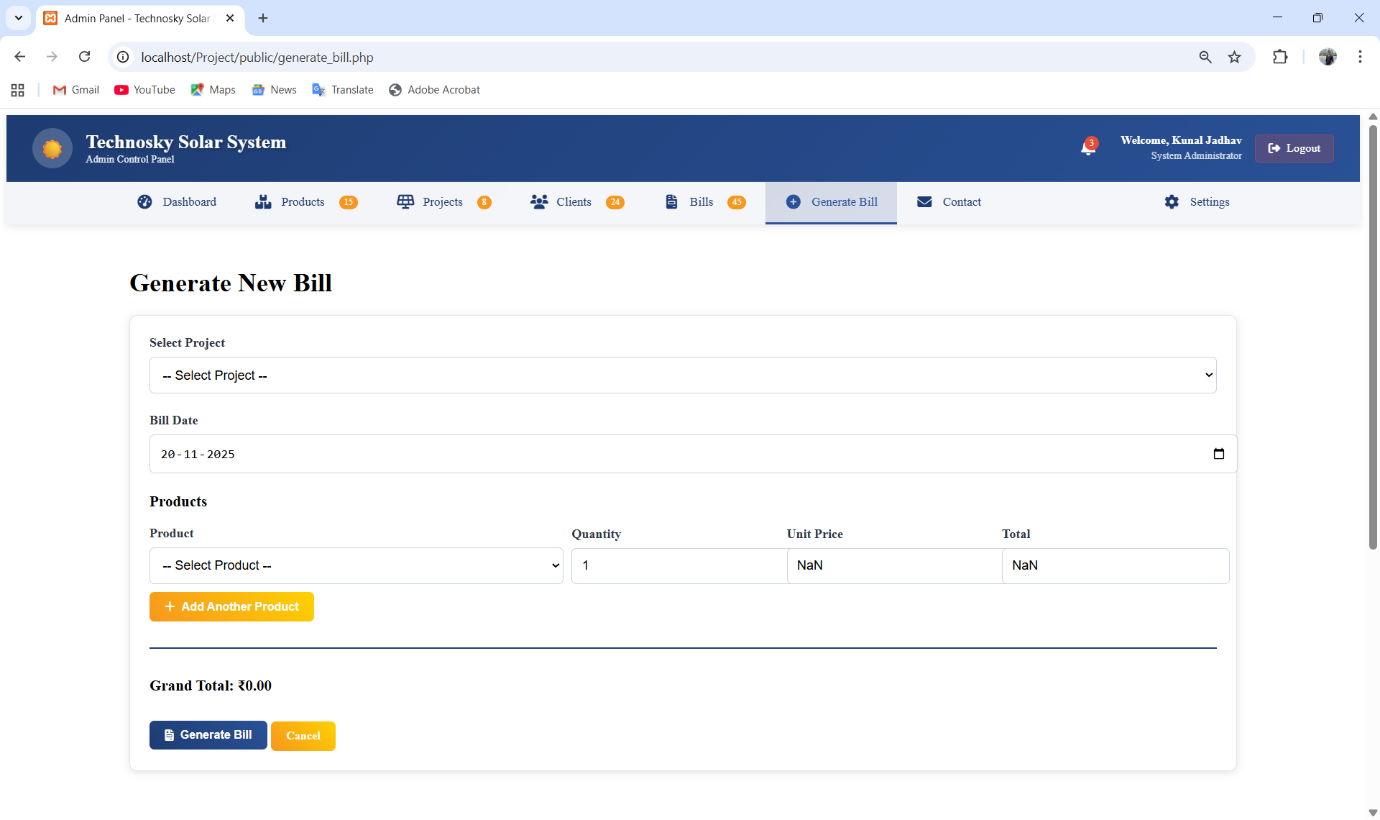
The Clients Management page is the CRM (Customer Relationship Management) hub of the application. It organizes customer data into a legible table format, displaying the Name, Email, Phone, and Address for every client. The screenshots show specific entries like "Chetan Lonkar" and "Priya Patel," demonstrating how the system handles different contact details and location formats (from specific "Girimphata" addresses to broader "Mumbai, Maharashtra" locations). This centralization of data is crucial for the sales and support teams to maintain contact.

**Footer**

****

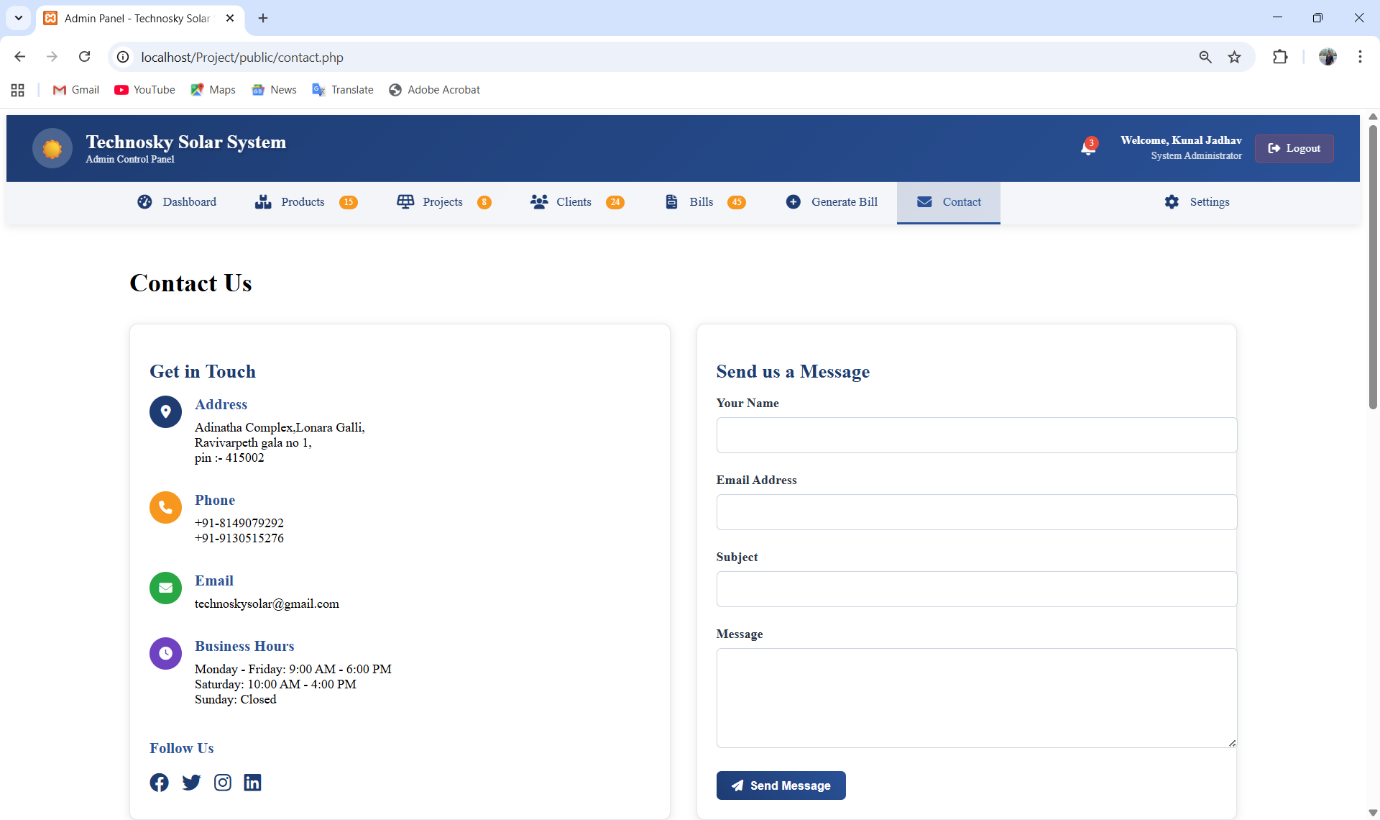
Below this sits the footer, which anchors the entire application. It is divided into four columns: a Brand column with the logo and a mission statement about sustainable energy; a Quick Links column for easy site navigation; a Contact Info column repeating the address and phone details; and a Business Hours column. A distinctive feature is the "24/7 Emergency Support Available" badge. The footer concludes with a copyright notice for 2024 Technosky Solar System and links to legal policies, giving the application a polished, enterprise-grade finish.

**Generate Bills**

****

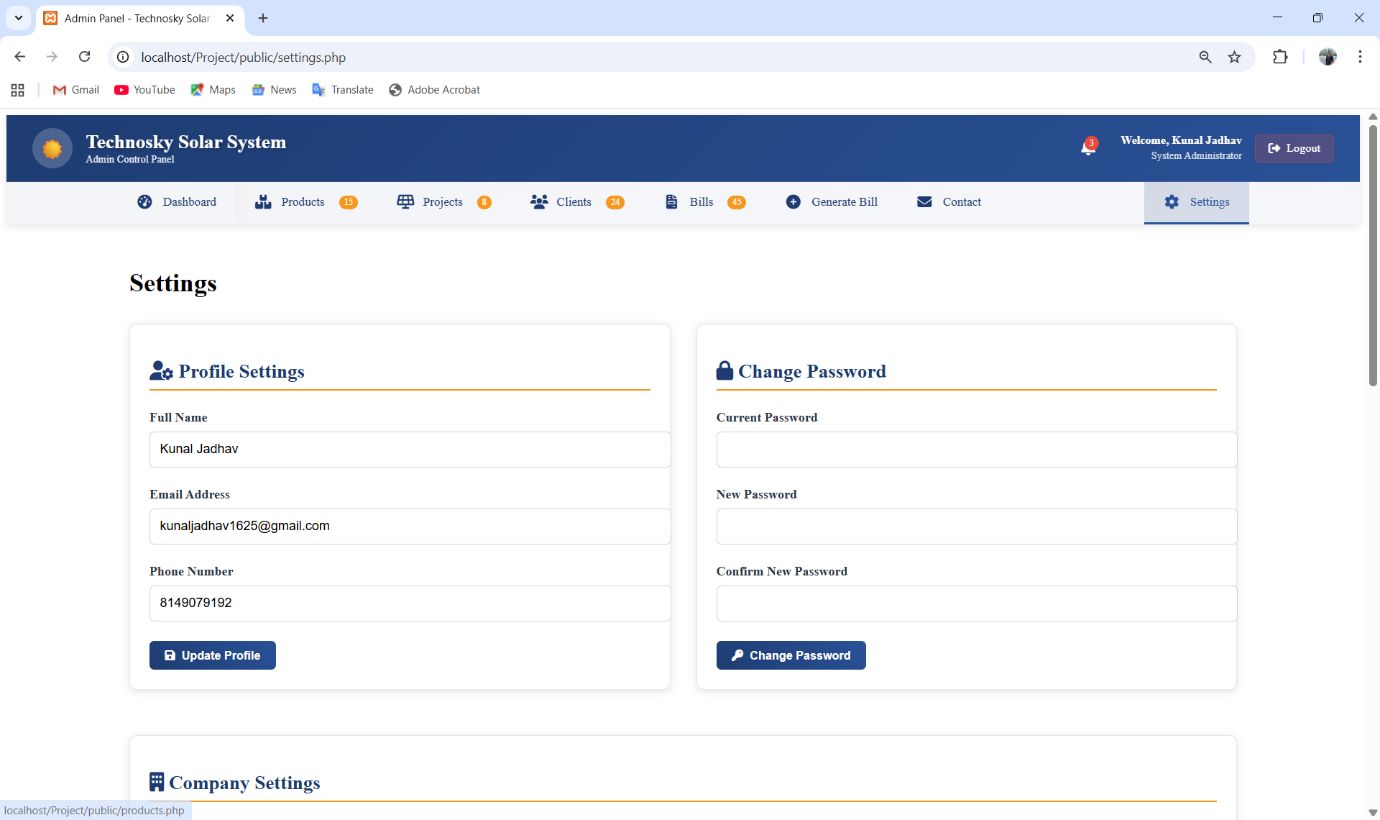
The Generate Bill page is a functional transaction interface designed to create invoices for clients efficiently. The layout is structured logically, starting with a Select Project dropdown menu, ensuring that every bill is accurately tied to an ongoing or completed job. Below this, a date picker allows the user to set the Bill Date, which defaults to the current date (20-11-2025). The core of this page is the dynamic "Products" section, where the administrator can itemize the services or hardware being sold.

**Contacts**

****

The Contact Page is designed with a dual-column layout to facilitate communication. The left column, titled Get in Touch, displays the company's static contact information, including the physical Address (Adinatha Complex, Satara), Phone numbers, Email address, and Business Hours. This serves as a reference for the admin or potentially as a preview of what clients see on the front end. It uses iconography (location pin, phone, envelope, clock) to make the information digestible.

**Settings**

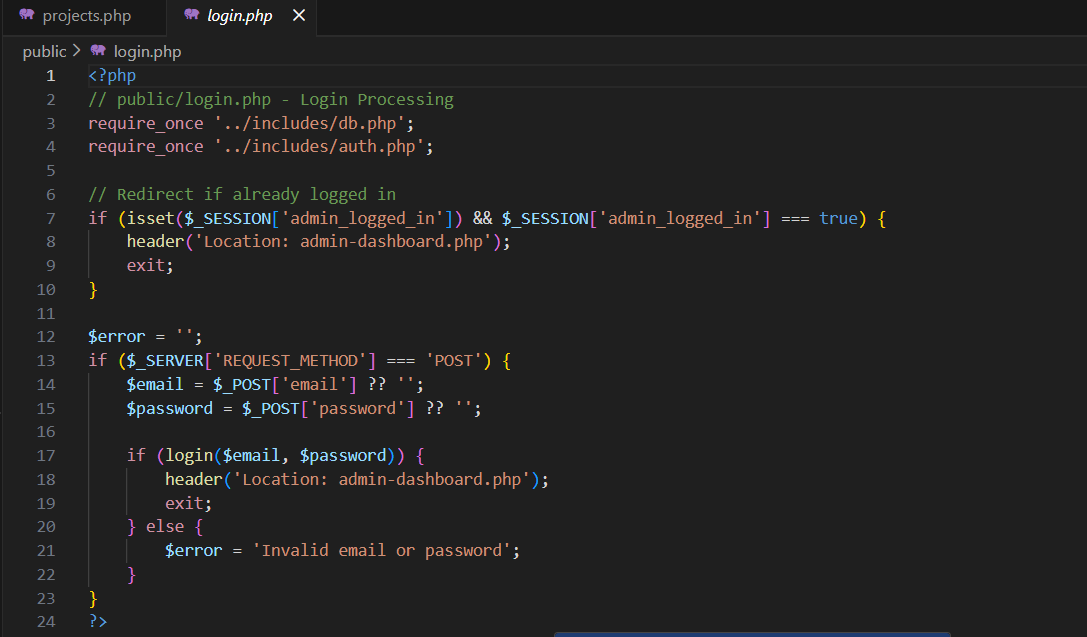
****

The Settings Page is the administrative control centre for user account management and system configuration. It features a clean, segmented layout using distinct "cards" to separate different categories of settings, ensuring a clutter-free user experience. The active tab in the navigation bar is clearly highlighted in grey, orienting the user within the application.

**Database Connection**

****

**Login**

****

**Authorization**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Advantages**

* Simple and easy-to-understand codebase – The system is built using plain PHP, making it easy to maintain and extend in the future.
* Secure and flexible database handling with PDO – PDO allows safe database operations using prepared statements and also supports switching between MySQL or PostgreSQL if needed.
* Well-organized folder structure – The use of includes/ for reusable code and public/ for page files keeps the project neat and manageable.
* Strong data consistency – Foreign keys and cascade rules in the database ensure that all linked records stay accurate and automatically update or delete when required.
* Centralized Data Management – All clients, products, projects, and bills are stored in one place, making it easy to access and manage records.
* Time-Saving and Efficient – Automates tasks like billing and inventory updates, reducing manual work.
* Easy to Use – Simple interface that even non-technical staff can operate without training.
* Accurate Billing – Generates itemized bills with automatic total calculation, avoiding human errors.
* Better Organization – Links clients, projects, and material usage clearly for future tracking.
* Improved Professionalism – Printed or digital bills look clean and structured, increasing customer trust.
* Secure Access – Only authorized users can log in, and passwords are stored securely.
* Error Reduction – Prepared statements and database validation prevent wrong entries and data corruption.
* Scalable System – Can be extended easily with more modules like roles, reports, or APIs in future.

**Future Enhancement**

While the current system is functional and reliable for handling basic administration tasks, there are several improvements that can be introduced to enhance scalability, performance, and security. The following enhancements would ensure that the system remains future-proof and capable of supporting more users, features, and enterprise-level operations.

One of the most critical improvements is implementing secure authentication mechanisms. The existing system can be upgraded by integrating *password hashing* using PHP’s password\_hash () and password\_verify () to protect user credentials. Furthermore, introducing Role-Based Access Control (RBAC) would allow different levels of access for super admins, billing operators, and regular staff.

In terms of security hardening, the system should adopt CSRF protection, form validation, and strict input sanitization across all modules to prevent injection attacks and unauthorized form submissions. For better maintainability, the project can be gradually migrated to a modern PHP MVC framework like Laravel or Symfony, which provides built-in support for routing, authentication, caching, and templating.

To support large-scale deployments, automated testing (PHPUnit) and Continuous Integration pipelines (GitHub Actions / GitLab CI) should be adopted. This ensures every update is tested before deployment. In the long term, the system can be extended to offer RESTful API endpoints so that it can be integrated with mobile apps or modern frontend frameworks like React or Vue. Additionally, maintaining activity logs and audit trails for billing and admin operations will improve accountability and tracking.

Recommended Improvements

* Implement password hashing using password\_hash / password\_verify.
* Add Role-Based Access Control (RBAC) for different admin levels.
* Apply CSRF protection and input sanitization across all forms.
* Migrate to Laravel / Symfony for better modularity and long-term maintenance.
* Add automated tests (PHPUnit) and CI/CD deployment pipelines.
* Provide REST API endpoints for mobile or frontend integration.
* Maintain detailed activity logs and audit trails for security monitoring.

**Conclusion**

The Technosky Solar System successfully establishes a structured foundation for managing clients, products, project workflows, and billing operations within a centralized platform. Its current implementation already demonstrates practical usability for small to medium-scale solar businesses, offering clear navigation and simplified record maintenance.

With a few essential security enhancements—such as enforcing password hashing, input validation, and HTTPS—the system can be safely deployed in real-world scenarios. Furthermore, modular refactoring into an MVC framework like Laravel or Symfony would significantly improve long-term maintainability, scalability, and developer collaboration.

Overall, this project provides not just a prototype but a solid baseline for a production-ready administrative system. It can be gradually extended into a fully-fledged ERP-like solution tailored specifically for solar industry service providers.