

ONLINE GAS BOOKING PROJECT IN PYTHON DJANGO

Sushree Upasana Pati, Ipsita Priyadarshini Dash, Prof. dr.Sujit Kumar Panda, *Computer Science And Engineering, Gandhi Institute For Technology, INDIA* sushree.pati2020@gift.edu.in

Abstract— The Online Gas Booking System Project in Python Django has been developed to override the problems prevailing in the practicing manual system. This Project is Supported to eliminate and in some cases reduce the hardship faced by this existing system. More over this Project is designed for the particular need of the company to carry out operations in a smooth and effective manner

(Users first fill the signup form then login into their panel and do the following activities)

- **Dashboard:** This is the Welcome page for the customer.
- **New Connection:** In this section, customer sends the request for LPG connection to the organization
- **Book Cylinder:** When the organization provides a connection number then the customer can book his/her cylinder.
- **Booking History:** In this section, customer can view the history of gas booking.
- **Search:** In this section, customer can search gas booking records by entering the booking number.
- The customer can also update his profile, change the password and recover the password.
- **Admin** is the superuser of the website who can manage everything on the website. Admin can log in through the login page
- **Dashboard:** In this section, admin can see all detail in brief like the total new connection, total new connection,
- total on-hold connection, total approved connection, total rejected connection, total new booking, total confirmed booking, total canceled booking, total assign booking, total delivered LPG, total staff and total registered users
- **Delivery Staff:** In this section, admin can manage staff (add/update).
- **Reg Users:** In this section, admin can view the detail of registered users.
- **Connection:** In this section, admin can view the connection request admin also has the right to change connection status according to the current status and add his/her remarks.
- **Booking:** In this section, admin can view booking request and assign to delivery staff or cancel the booking.
- **Assigned Booking:** In this section, admin can change the status of booking according to the current status and add his/her remarks.
- **Reports:** In this section, admin can view booking and connection requests in a particular period.

KEYWORDS— “LPG REFILL BOOKING”, “ONLINE LPG BOOKING”, “BOOK GAS CYLINDER ONLINE”, “REFILL GAS ONLINE”, “GAS AGENCY ONLINE BOOKING”, “ONLINE GAS DELIVERY”.

1.1 INTRODUCTION

In today's fast-paced world, where convenience is paramount, traditional methods of gas booking are undergoing a transformative shift towards digitization. The advent of online platforms has revolutionized various industries, and the gas distribution sector is no exception. Our project embarks on the development of an Online Gas Booking System using Python Django, a robust web framework known for its scalability and efficiency. This system aims to bridge the gap between traditional gas booking methods and contemporary consumer expectations by offering a user-friendly, accessible, and efficient platform.

Key Points:

Digital Transformation: Transitioning from manual to digital processes for gas booking, our project aims to offer users a seamless and convenient online platform.

Python Django Framework: Leveraging the power of Python Django, we're poised to develop a feature-rich and scalable solution for gas booking, ensuring reliability and performance.

Enhanced User Experience: Prioritizing user experience, our platform will offer intuitive navigation, clear instructions, and responsive design for effortless gas booking.

Operational Efficiency: By streamlining operations for both users and gas agencies, our project seeks to enhance efficiency and optimize resource allocation.

Objective:

Our objective is to redefine the gas booking experience by developing a user-centric, tech-savvy solution that meets the evolving needs of users and gas agencies alike.

1.2 Problem Statement

Problem Statement: Online Gas Booking Project in Python Django

In today's digital age, traditional methods of gas booking often involve cumbersome processes, long wait times, and limited accessibility. Users face challenges such as:

Inconvenience: Users often have to visit gas agencies in person or wait in long queues to book gas cylinders, leading to time wastage and frustration.

Limited Accessibility: Those in remote or underserved areas may face difficulties accessing gas booking services due to geographical constraints or lack of internet connectivity.

Security Concerns: Traditional payment methods may pose security risks, with users hesitant to share sensitive financial information over the phone or in person.

Lack of Transparency: Users may not have real-time visibility into the status of their gas bookings, leading to uncertainty and frustration.

Inefficient Operations: Gas agencies may struggle with manual processes, leading to inefficiencies in order processing, delivery scheduling, and inventory management.

Objective:

The objective of the Online Gas Booking Project in Python Django is to address these challenges by developing a comprehensive online platform that offers users a convenient, secure, and transparent way to book gas cylinders.

Through the implementation of a user-friendly interface, secure payment integration, real-time order tracking, and efficient backend operations, the project aims to streamline the gas booking process for users and gas agencies alike, enhancing overall satisfaction and operational efficiency.

1.3 Suggested Solutions

Suggested Solutions for Online Gas Booking Project in Python Django:

User-Friendly Interface:

Design a clean and intuitive user interface that guides users through the gas booking process seamlessly, ensuring ease of use and accessibility for all users.

Secure Authentication and Authorization:

Implement robust authentication mechanisms such as OAuth or JWT tokens to ensure secure access to the platform and protect user data.

Responsive Web Design:

Develop a responsive web design that adapts to different screen sizes and devices, providing a consistent user experience across desktops, tablets, and smartphones.

Payment Gateway Integration:

Integrate trusted payment gateways such as PayPal, Stripe, or Razorpay to facilitate secure online transactions, ensuring the safety of user financial information.

Real-Time Order Tracking:

Implement a real-time order tracking system that allows users to track the status of their gas bookings from placement to delivery, providing transparency and peace of mind.

Notification System:

Set up a notification system to send automated email or SMS notifications to users at various stages of the booking process, keeping them informed and updated.

Admin Dashboard:

Develop an admin dashboard for gas agency administrators to manage orders, track deliveries, view customer data, and monitor inventory levels efficiently.

Inventory Management:

Build an inventory management system that automatically updates stock levels in real-time as orders are placed and fulfilled, preventing overbooking or stockouts.

Customer Support Integration:

Integrate customer support features such as live chat, helpdesk ticketing systems, or chatbots to provide timely assistance and resolve user queries effectively.

Data Analytics and Reporting:

Implement data analytics tools to gather insights into user behavior, booking trends, and operational performance, enabling informed decision-making and optimization strategies.

1.4 Objectives**Enhanced Convenience:**

Develop a user-friendly platform that enables users to book gas cylinders conveniently from anywhere, reducing the need for physical visits to gas agencies.

Efficient Booking Process:

Implement a streamlined gas booking process that guides users through selecting cylinder types, specifying delivery details, and completing transactions efficiently.

Seamless Booking Experience: Develop a user-friendly interface that simplifies the gas booking process, allowing users to place orders quickly and efficiently. **Real-Time Order Tracking:** Enable users to track the status of their gas bookings in real-time, providing transparency and assurance regarding delivery timelines.

Efficient Management for Gas Agencies:

Provide gas agencies with tools to manage orders, track deliveries, and optimize inventory levels for efficient operations.

User Account Management: Allow users to create accounts, manage their profiles, and view their booking history for easy reference and tracking. **Customer Support Integration:** Integrate customer support features such as chat support or email assistance to address user queries and concerns promptly.

1.5 Needs of Online Gas Booking

Convenience: Users require a convenient way to book gas cylinders without the need for physical visits to gas agencies, saving time and effort. **Accessibility:** Users need access to gas booking services regardless of their location or time constraints, especially in remote or underserved areas.

Transparency: Users seek transparency in the gas booking process, including real-time updates on order status, delivery timelines, and payment transactions. **Security:** Users require secure payment options and data protection measures to ensure the safety of their financial information and personal data.

1.6 Functionalities

These are the functionality performed by the Project.

- User Registration and Authentication:
- Gas Cylinder Selection
- Delivery Details
 - Real-time Availability Check
 - Booking Confirmation
 - Payment Integration

- Order Tracking
- Delivery Notifications

1.6.1 Dashboard for Admin User

- Order Management
- Delivery Tracking
- Inventory Management
- User Management
- Security and Access Control

1.6.2 Manage Gas Booking Details

- View Booking Details
- Update Booking Status
- Reschedule or Cancel Bookings

1.6.3 Manage Booking Cart

- Add Gas Cylinders to Cart
- Edit Cart Items
- View Cart Summary
- Calculate Total Cost

1.6.4 Manage Customer Registration

- Adding New Customer Registration
- Edit the Existing Customer Registration
- View details of the Customer Registration

1.6.5 Manage Confirm Booking

- Adding New Confirm Booking
- Edit the Existing Confirm Booking
- View details of the Confirm Booking

1.7 Features

Features of Online Gas Booking System:

User Registration and Authentication:

Allow users to register for an account securely and authenticate themselves to access booking features.

Gas Cylinder Selection:

Provide a variety of gas cylinder options for users to select based on their preferences and requirements.

Delivery Scheduling:

Enable users to schedule gas cylinder deliveries at their preferred date and time, providing flexibility and convenience.

Real-time Availability Check:

Check the real-time availability of gas cylinders and delivery slots to ensure timely fulfillment of orders.

Secure Payment Integration:

Integrate secure payment gateways to facilitate safe and convenient online transactions for booking payments.

Order Tracking and Notifications:

Allow users to track the status of their orders in real-time and receive notifications for order confirmation, dispatch, and delivery.

User Profile Management:

Provide users with the ability to manage their profiles, update personal information, and view booking history.

Admin Dashboard:

Offer gas agency administrators a centralized dashboard to manage bookings, track deliveries, and monitor inventory levels.

Inventory Management:

Enable gas agencies to manage gas cylinder inventory, track stock levels, and ensure availability for bookings.

Customer Support Integration:

Integrate customer support features such as live chat or email support to assist users with queries and issues.

Responsive Design:

Ensure the platform is responsive and accessible across various devices and screen sizes for a seamless user experience.

Feedback Mechanism:

Incorporate a feedback mechanism for users to provide input and suggestions for platform improvement.

Localization and Internationalization:

Support multiple languages and currencies to cater to users from different regions and enhance accessibility.

Security Measures:

Implement robust security measures to protect user data and ensure secure transactions, including encryption and secure authentication.

Scalability and Performance Optimization:

Design the platform to handle a large number of users and transactions efficiently, optimizing performance and reliability.

These features collectively aim to provide users with a comprehensive, secure, and user-friendly online gas booking experience while enabling gas agencies to manage operations effectively.

LITERATURE REVIEW**2.1 *Background of the Studies***

The transition from traditional, manual methods of gas booking to online platforms marks a significant shift in the gas distribution industry. Historically, gas booking has been a cumbersome process, often involving physical visits to gas agencies, long wait times, and limited accessibility. However, with the advent of digital technology and the proliferation of internet connectivity, online gas booking systems have emerged as a viable solution to address these challenges.

The background of this study lies in the recognition of the need for a more convenient, efficient, and accessible method of gas booking. Traditional methods were fraught with inefficiencies and lacked transparency, leading to frustrations among consumers and operational challenges for gas agencies. Moreover, the rapid pace of technological advancement and changing consumer preferences necessitated the adoption of digital solutions to stay competitive in the market.

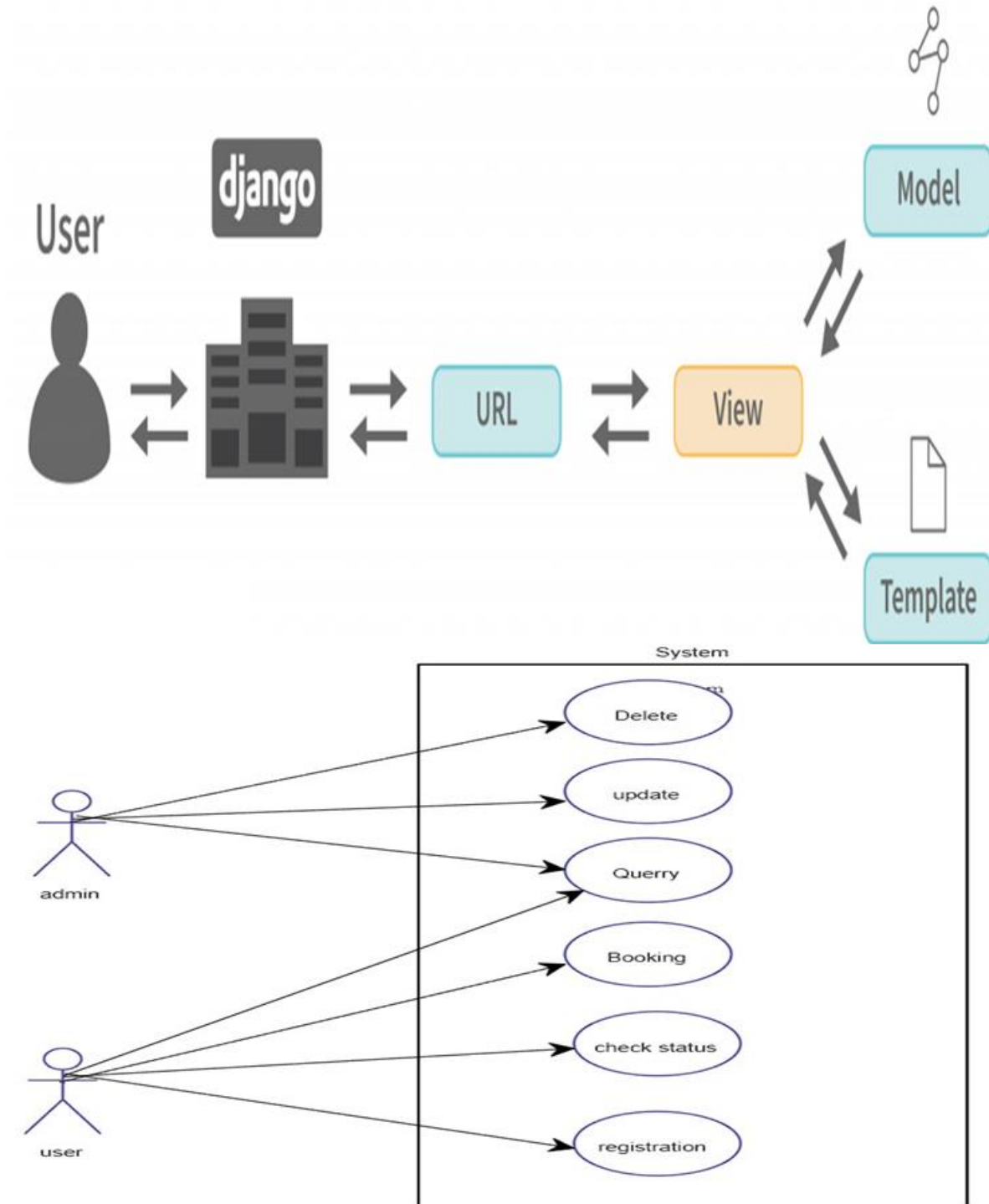
Against this backdrop, the development of an online gas booking system using Python Django represents a proactive response to the evolving needs of both consumers and gas agencies. By leveraging the power of technology, this study aims to streamline the gas booking process, enhance user experience, and optimize operational efficiency for gas agencies.

The background of the study also encompasses a review of existing literature, market trends, and technological advancements in the field of online gas booking. By synthesizing insights from previous research and industry developments, this study seeks to identify best practices, challenges, and opportunities in the design and implementation of online gas booking systems.

In summary, the background of this study underscores the importance of embracing digital transformation in the gas distribution industry to meet the demands of modern consumers and drive operational excellence. Through the development of an online gas booking system, this study endeavors to contribute to the advancement of digital solutions in the gas sector and enhance overall service delivery for consumers and gas agencies alike.

METHODOLOGY**3.1 *Complete Visualization Model***

The complete visualization model for the online gas booking system encompasses various aspects of the platform's functionality and user interface. It includes:



3.2 **Hardware Requirement**

- Server
- Database Server
- Backup and Disaster Recovery
- Monitoring and Management Tools

3.3 **Software Requirement**

- Operating System:
 - For server infrastructure: Linux distributions (e.g., Ubuntu, CentOS) or Windows Server
 - For client devices: Windows, macOS, Linux, Android, iOS
- Web Server:
 - Apache HTTP Server, Nginx, or Microsoft Internet Information Services (IIS) for hosting the web application
- Database Management System (DBMS):
 - MySQL, PostgreSQL, or SQLite for storing and managing application data

- Backend Framework:
- Django: A high-level Python web framework for rapid development of web applications with built-in features like ORM, authentication, and URL routing
- Frontend Technologies:
- HTML5, CSS3, JavaScript (ES6+), and responsive design frameworks (e.g., Bootstrap, Materialize) for building the user interface of the web application
- Vue.js, React, or Angular for developing dynamic and interactive frontend components (optional)
- Version Control:
- Git for managing source code repositories, collaboration, and version control
- Integrated Development Environment (IDE):
- PyCharm, Visual Studio Code, or Sublime Text for Python development
- Android Studio or Xcode for mobile app development
- Software Requirements:
- Operating System:
- Server: Linux-based operating systems such as Ubuntu Server, CentOS, or Red Hat Enterprise Linux (RHEL) are commonly used for hosting web applications.
- Development Environment: Any operating system compatible with Python Django development tools, including Windows, macOS, and Linux distributions.
- Web Server:
- Apache HTTP Server or Nginx: Popular web server software for hosting Django applications and serving web pages to clients.
- Database Management System (DBMS):
- PostgreSQL: A robust open-source relational database management system (RDBMS) known for its reliability, scalability, and support for complex queries.
- MySQL: Another popular open-source RDBMS with strong community support and compatibility with Django applications.
- SQLite: Lightweight and easy-to-use database engine suitable for development and testing environments.
- Python and Django Framework:
- Python: Version 3.x (e.g., 3.6, 3.7, 3.8) as the programming language for developing the application logic and backend functionality.
- Django Framework: The latest stable version of Django framework (e.g., Django 3.x) for building web applications quickly and efficiently.
- Dependency Management:
- pip: The package installer for Python used to install and manage dependencies required by the Django application.
- Frontend Technologies:
- HTML, CSS, JavaScript: Standard web development technologies for building the user interface (UI) and frontend components of the application.
- Bootstrap or Materialize CSS: Frontend frameworks for designing responsive and visually appealing UI elements.
- Development Tools:
- Integrated Development Environment (IDE): Use an IDE such as PyCharm, Visual Studio Code, or Sublime Text for Django application development, offering features like code completion, debugging, and version control integration.
- Version Control: Git for version control management, allowing collaborative development and tracking of code changes.
- Security Tools and Libraries:
- Django Security Middleware: Built-in security features provided by Django, including protection against common web vulnerabilities such as Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF).
- SSL/TLS Certificates: Secure Socket Layer (SSL) or Transport Layer Security (TLS) certificates for encrypting data transmitted between clients and the server.
- Deployment and Hosting:

- **Deployment Platform:** Choose a deployment platform such as Heroku, AWS Elastic Beanstalk, or DigitalOcean Droplets for hosting the Django application in a production environment.

3.4 Tools and Technique

3.4.1 Django

Django is a versatile web framework known for its efficiency in creating robust and maintainable web applications. Developed in Python, Django promotes rapid development through its "batteries-included" approach, offering a wide range of built-in features such as authentication, URL routing, and database management. Its adherence to the Model-View-Template (MVT) architectural pattern enhances code organization and scalability. Additionally, Django prioritizes security with built-in protections against common web vulnerabilities and a robust authentication system. Its object-relational mapping (ORM) layer simplifies database interactions, while its vibrant community provides extensive support, documentation, and third-party packages. Overall, Django is valued for its simplicity, security, and thriving ecosystem, making it a preferred choice for developers tackling web development projects of any size or complexity.

3.4.2 MySQL

MySQL Workbench is a comprehensive visual tool for DBAs, database architects, and developers. Data modeling, SQL creation, and extensive administrative tools for server configuration, user management, backup, and other tasks are all provided by MySQL Workbench. There are versions of MySQL Workbench for Windows, Linux, and Mac OS.

3.4.3 HTML

Hypertext Markup Language (HTML) is the industry-standard markup language for developing web apps and pages. It is one of three foundational technologies underpinning the World Wide Web, along with JavaScript and Cascading Style Sheets (CSS). HTML documents are downloaded from a web server or local storage by web browsers, who then turn them into multimedia web pages. HTML originally featured cues for the document's design and semantically explains the structure of a web page. The foundation of HTML pages are HTML components. Images and other objects, like interactive forms, may be embedded within the produced page using HTML techniques. By indicating structural semantics for text elements like headings, paragraphs, lists, links, quotations, and other objects, HTML offers a way to generate structured texts.

3.4.4 Bootstrap

Bootstrap is a front-end framework that is open-source and free to use while creating websites and web apps. It includes optional JavaScript extensions along with HTML and CSS-based design templates for navigation, buttons, forms, buttons, and other interface elements. It only addresses front-end development, unlike many web frameworks.

3.4.5 Visual Studio

Sublime Text is a commercial cross-platform source code editor that utilizes the Python programming language (API). Numerous programming and markup languages are supported natively, and users can add features through plugins, which are often developed and maintained by the local community under free-software licenses.

3.4.6 Git hub

GitHub is a Git-based version control hosting service on the internet. Code is where it is most frequently utilized. It has all of Git's distributed version control and source code management (SCM) features in addition to a few extras. Every project can benefit from access control and a variety of collaborative tools, including wikis, task management, issue tracking, and feature requests. Both private repositories and free accounts, which are frequently used to host open-source software projects, are available on GitHub.

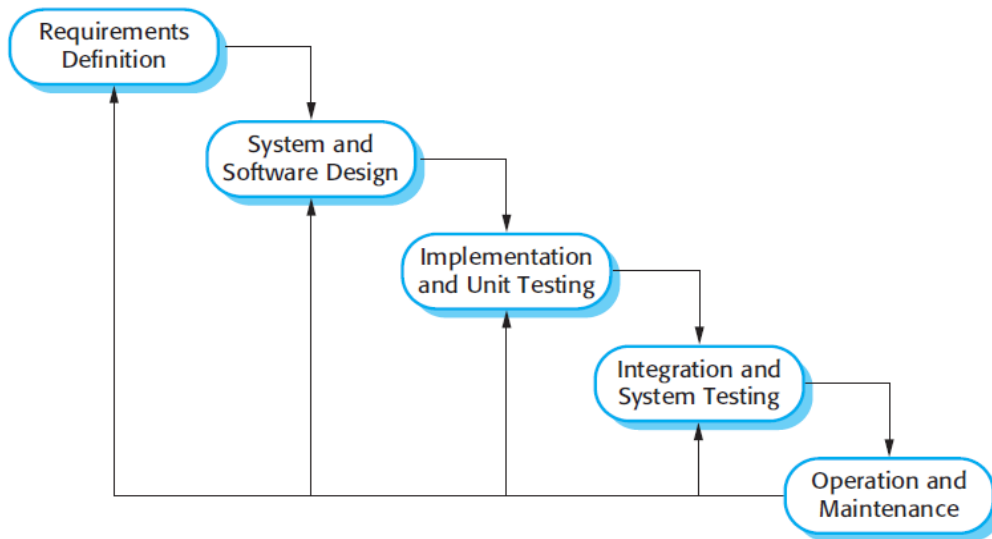
3.4.7 Java Script

JavaScript often abbreviated as JS, is an interpreted, high-level programming language. Additionally, it is a dynamic, weakly typed, prototype-based, and multi-paradigm language. One of the three fundamental technologies of the World Wide Web, together with HTML and CSS, is JavaScript. JavaScript is a crucial component of online applications because it makes web pages interactive. The vast majority of websites make use of it, and every significant web browser has an engine specifically designed to run JavaScript.

3.4.8 CSS

Cascading Style Sheets (CSS) is a language for creating style sheets that describe how a document produced in a markup language like HTML will look. The World Wide Web's foundational technologies, along with HTML and JavaScript, include CSS. Layout, color, and font may all be separated from content and presentation using CSS. By describing the pertinent CSS in a separate CSS file, this separation can make content more accessible, give definition of presentation features greater freedom and control, allow numerous web pages to share formatting, and reduce complexity and repetition in structural content.

3.5 Methodology Development Model



The Waterfall model's consecutive phases are:

Requirement Gathering and analysis – During this stage, all potential system needs are gathered and outlined in a requirement specification document.

- **System Design** – The system design is created in this phase after studying the requirement specifications from the first phase. This system design aids in determining the overall system architecture as well as the hardware and system requirements.
- **Implementation** – The system is initially built in discrete programs known as units, which are then combined in the following phase, using inputs from the system design. Unit testing is the process of developing and evaluating each unit for functionality.
- **Integration and Testing** – Following the testing of each unit created during the implementation phase, the entire system is merged. The entire system is tested for errors and failures after integration.
- **Deployment of system** – Once the product has undergone functional and non-functional testing, it is either published to the market or deployed in the customer's environment.
- **Maintenance** – Various problems can arise in a client environment. Patches are published to address certain problems. Additionally, improved versions of the product are issued. To bring about these changes in the surroundings of the consumer, maintenance is performed.

3.6 System Design Model

Home Page



CONCLUSION & FUTURE WORK

Conclusion

The implementation of an online gas booking system presents a significant opportunity to revolutionize the way gas cylinders are booked and managed. Throughout this analysis, several key points have emerged:

Enhanced Convenience: By providing users with a convenient and accessible platform for booking gas cylinders, the online booking system streamlines the booking process and saves users valuable time and effort.

Improved Efficiency: The online booking system automates many aspects of the booking process, reducing manual paperwork and administrative overhead for gas agencies, and improving overall operational efficiency.

Transparency and Accountability: With features such as transparent pricing, real-time order tracking, and automated reminders, the online booking system fosters greater transparency and accountability in the gas distribution process.

Customer Satisfaction: By offering a seamless and user-friendly booking experience, the online booking system enhances customer satisfaction and loyalty, driving positive word-of-mouth and repeat business.

Challenges and Limitations: It's important to acknowledge the challenges and limitations associated with implementing an online gas booking system, such as connectivity issues, security concerns, and user resistance to change. However, with careful planning and mitigation strategies, these challenges can be overcome.

In conclusion, the implementation of an online gas booking system represents a transformative step towards modernizing the gas distribution industry and meeting the evolving needs of customers. By leveraging the advantages of digital technology, gas agencies can improve efficiency, enhance customer satisfaction, and drive sustainable growth in the long term.

5.1 Future Work

Mobile Application Development

Consider developing a dedicated mobile application for the online gas booking system to cater to users who prefer to access services on their smartphones or tablets. This would enhance accessibility and convenience, potentially increasing user adoption rates.

5.2.2 Integration with Smart Home Devices

Explore the possibility of integrating the online gas booking system with smart home devices, such as IoT-enabled gas meters or voice-activated assistants. This could enable seamless reordering of gas cylinders based on consumption patterns, further enhancing user convenience.

5.2.3 Expansion of Service Offerings

Consider expanding the service offerings of the online gas booking system beyond gas cylinder booking. This could include additional services such as appliance maintenance scheduling, gas safety checks, or even home delivery of other household essentials.

5.2.4 Enhanced Analytics and Insights

Invest in advanced analytics tools to gain deeper insights into user behavior, booking patterns, and market trends. This data-driven approach can inform strategic decision-making, improve service offerings, and drive business growth.

5.2.5 Integration with Payment Wallets

Integrate popular payment wallets or digital payment platforms into the online gas booking system to offer users more flexibility and convenience in making transactions. This could also help streamline the payment process and reduce friction for users.

5.2.6 Expansion to New Markets

Explore opportunities to expand the online gas booking system to new geographic markets or customer segments. Conduct market research and feasibility studies to identify potential growth areas and tailor the system to meet the unique needs of diverse audiences.

5.2.7 Continuous Improvement and Feedback Mechanisms

Implement mechanisms for collecting user feedback and suggestions for improvement on an ongoing basis. This feedback can be used to iteratively refine the system, address pain points, and ensure that it remains relevant and competitive in the market.

5.2.8 Partnerships and Collaborations

Explore potential partnerships or collaborations with other stakeholders in the gas distribution ecosystem, such as gas suppliers, delivery services, or smart home technology providers. By leveraging synergies and complementary strengths, you can enhance the value proposition of the online gas booking system and drive mutual benefits.

5.2.9 Regulatory Compliance and Sustainability Initiatives

Stay abreast of regulatory changes and industry standards related to online transactions, data privacy, and environmental sustainability. Ensure that the online gas booking system remains compliant with relevant regulations and adopts sustainable practices to minimize its environmental footprint.

References

Gleditsch, N. P., Pinker, S., Thayer, B. A., Levy, J. S., & Thompson, W. R. (2013). The forum:

- [1] Data Analysis. International Data Analysis Review, 15(3), 396-419.
- [2] Google for problem solving.
- [3] <https://www.slideshare.net/>
- [4] Database Programming with JDBC and Java by O'Reilly
- [5] <http://www.jdbc-tutorial.com/>
- [6] <https://www.tutorialspoint.com/java/http://www.javatpoint.com/java-tutorial>
- [7] <https://docs.oracle.com/javase/tutorial/>
- [8] <http://www.wampserver.com/en/>
- [9] <http://www.JSP.net/>
- [10] <http://www.tutorialspoint.com/mysql/>