A PROJECT REPORT ON  
Import-Export Business Website

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE BACHELOR OF COMPUTER APPLICATION (BCA)

SUBMITTED BY  
Yagni Rajesh Mehta



UNDER THE GUIDANCE OF  
MS. VAIDEHI RAVAL

JANSEVA SAMITI SANCHALIT  
SHRI. M. D. SHAH MAHILA COLLEGE OF ARTS AND COMMERCE  
(AFFILIATED COLLEGE TO S.N.D.T WOMEN’S UNIVERSITY)  
B.J. PATEL ROAD MALAD (WEST) MUMBAI-400064  
2024-2025

DECLARATION

I, Ms. Yagni Rajesh Mehta, a student of Bachelor of Computer Applications (BCA) Semester VI (2024-2025), hereby declare that the Project Report submitted to Shri M. D. Shah Mahila College has been entirely developed by me to fulfil the requirements of the final year project. I affirm that this project has been completed within the specified time frame. The information provided by me is true and original to the best of my knowledge.

Yagni Rajesh Mehta - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ACKNOWLEDGEMENT

I started this project as a part of the curriculum of B.C.A., assigned to me by "SHRI M.D. SHAH MAHILA COLLEGE". It was very exciting for me to work on my project for Reliable Exim.

I take this opportunity to express my heartfelt gratitude to all the people from my college for inspiring me to gain knowledge of the subject through theoretical and practical approaches at every stage of my training.

I am sincerely indebted to Ms. Vaidehi Raval, who extended valuable guidance, constant encouragement, and helped through various difficult stages for the completion of my project.

I owe my deep gratitude and respect to the College Management and respected Principal Dr. Deepa Sharma for encouragement and facilities provided to me.

I would also like to thank my faculty members without whom this project would have been a distant reality. In the end, I would like to thank my family and friends for their positive criticism and needful support.

CLIENT CERTIFICATE

INDEX

| **SR NO.** | **CONTENTS** | **PAGE NO.** |
| --- | --- | --- |
| 1. Synopsis | 1.1 About Import-Export Business Website |  |
|  | 1.2 Project Introduction |  |
|  | 1.3 Technical Specification |  |
|  | 1.4 Beneficiaries |  |
| 2. SDLC | 2. System Development Life Cycle (SDLC) - Model Followed |  |
| 3. System Planning | 3.1 Gantt Chart |  |
| 4. System Analysis | 4.1 Requirements Gathering |  |
|  | 4.1.1 Functional Requirements |  |
|  | 4.1.2 Non-Functional Requirements |  |
|  | 4.2 Modal |  |
|  | 4.3 Feasibility Study |  |
|  | 4.4 Software Requirements Specification |  |
|  | 4.5 Tools & Technology |  |
| 5. System Design | 5.1 Module Decomposition Tree |  |
|  | 5.2 Flowchart |  |
|  | 5.3 ER diagram |  |
|  | 5.4 UI Screenshots and Wireframes |  |
| 6. System Coding | 6.1 Code |  |
| 7. System Testing | 7.1 Black box testing |  |
|  | 7.2 White box testing |  |
| 8. Implementation | 8.1 User Guide |  |
|  | 8.2 Hosting Details |  |
|  | 8.3 User Guide |  |
| 9. Conclusion | 9.1 Lessons of Learning |  |
| 10. | 10.1 Future Scope / Enhancements |  |
| 11. | 11.1 List of References - Bibliography, Webliography |  |

# 1.SYNOPSIS

1.1 About Import-Export Business Website  
The project aims to create a dynamic import-export business website that enhances customer engagement, provides real-time shipment tracking, and integrates WhatsApp for instant communication. The system will also feature an admin dashboard for business insights and SEO optimization to improve visibility.

1.2 Project Introduction

**Problem with Existing Systems**

* Traditional import-export websites lack automation, leading to manual and time-consuming order processing.
* Poor tracking mechanisms make it difficult for customers to monitor their shipments in real time.
* Communication between businesses and customers is often slow and inefficient, with no real-time support options.
* Limited scalability makes it challenging to handle an increasing number of transactions and customers.
* Security concerns due to lack of encryption and authentication mechanisms, leading to potential data breaches.
* Lack of interactive UI/UX, making it difficult for users to navigate and use the website effectively.
* Lack of integration with third-party services, reducing the efficiency of managing logistics and trade.

**Solution Offered**

* Automated order processing to eliminate manual work and speed up transactions.
* Real-time shipment tracking system using QR codes and barcodes for better visibility.
* WhatsApp and chatbot integration to provide instant customer support and inquiries.
* Scalable and secure architecture, ensuring the website can handle high traffic and transactions.
* Admin dashboard with analytics, helping business owners track customer engagement, inquiries, and conversions.
* SEO-optimized platform, ensuring better online visibility and increased customer acquisition.
* Enhanced UI/UX design, improving user experience and increasing customer retention.
* Integration with third-party logistics services, ensuring smooth coordination between businesses and shippers.

1.3 Technical Specification

**Frontend (User Interface)**

* HTML, CSS, JavaScript (GSAP/Three.js for animations)
* WhatsApp API (for direct message redirection)

**Backend (Processing & Data Storage)**

* Django (backend framework)
* Django REST Framework (DRF) (for tracking API)
* PostgreSQL (database for storing orders & tracking)

**Tracking System**

* Barcode / QR Code Library (qrcode, pyzbar) (for tracking IDs)
* RFID/barcode scanners (to update tracking info automatically)

**Deployment**

* DigitalOcean / AWS (to host the backend)
* Gunicorn + Nginx (to handle high traffic)

**System Requirements:**  
*Minimum Requirements to Run the System:*

* **Operating System:** Windows 10/11, macOS, Linux
* **Processor:** Intel i3 or equivalent (minimum)
* **RAM:** 4GB (minimum), 8GB (recommended)
* **Storage:** 10GB free space
* **Browser Compatibility:** Chrome, Firefox, Edge

1.4 Beneficiaries

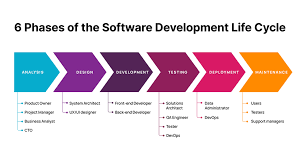
**Target Users**

* Import-export businesses looking to automate and optimize trade processes.
* Freight and logistics companies that require real-time shipment tracking.
* Business owners needing an interactive, scalable, and secure trade management system.
* Customers who need easy access to shipment tracking and real-time updates.
* E-commerce traders looking for a seamless solution to manage bulk international orders.
* Government agencies that monitor trade compliance and shipping logistics.

**Advantages**

* Enhances customer engagement and satisfaction through real-time tracking.
* Automates and streamlines order management, reducing manual errors.
* Provides insights and analytics for better business decision-making.
* Improves communication efficiency with integrated chat and chatbot features.
* Boosts visibility through SEO optimization, attracting more customers.
* Ensures high security and encryption, protecting business transactions and customer data.

# 2.SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)



2.1 Introduction to SDLC  
The software development lifecycle (SDLC) refers to the process that IT teams employ to design, develop, test, deploy, and maintain high-quality software applications and systems. It is crucial to understand the phases of SDLC for managing large or complex software projects.

2.2 Phases of SDLC

Here is a brief overview of the software development life cycle:

1. **Project Planning**
2. **Analysis of the Project**
3. **Project Design**
4. **Implementation of the Project**
5. **Project Testing**
6. **Project Deployment**

2.3 Importance of SDLC

SDLC is important because it provides a framework for developing quality software applications in a timely and cost-effective manner. It also helps ensure that the application meets the user’s needs and requirements. By adhering to it, organizations can reduce their risks and gain visibility into the development process.

Organizations frequently use the System Development Life Cycle to create and maintain software applications and other associated systems. Planning, designing, creating, testing, and deploying software applications is a systematic process.

2.4 How SDLC Works?

The process known as the Software Development Life Cycle (SDLC) is commonly used by software development teams to plan, create, test, and launch software products. Here’s a breakdown of how it unfolds:

1. **Planning:** The initial phase where the project scope, objectives, requirements, and resources are defined. The key players collaborate to craft a project blueprint and set timelines.
2. **Analysis:** This stage involves examining requirements by team members, including stakeholders, to understand user needs, business goals, and technical constraints.
3. **Design:** The team drafts the software design based on the analyzed requirements. This includes making plans, creating database structures, and designing user interfaces.
4. **Implementation:** Developers write code in compliance with design specifications. This covers constructing software components and integrating them.
5. **Testing:** The software application undergoes critical testing to identify any errors that may have occurred during development. Testing includes unit testing, integration testing, system testing, and acceptance testing to ensure compliance with quality standards.
6. **Deployment:** Once tested and approved, the software is rolled out into the production environment. This involves installation, configuration, and making the software accessible to users.
7. **Maintenance:** Post-deployment, the focus shifts to incorporating user feedback, resolving issues, introducing functionalities, and ensuring the software continues to meet evolving needs.

2.5 Benefits of SDLC

**2.5.1 Enhanced Visibility and Collaboration**

The SDLC framework facilitates alignment among stakeholders and team members by establishing objectives, deliverables, and milestones for each phase. This increased transparency enables managers, developers, and customers to collaborate effectively in launching high-quality software.

**2.5.2 Improved Estimation, Planning, and Tracking**

Breaking down projects into SDLC phases enhances estimation accuracy, schedule management, resource allocation, and overall project workflow. Monitoring progress ensures the development process stays on track in terms of both time and budget.

**2.5.3 Mitigated Risks**

The early gathering of requirements and regular testing help prevent costly scoping issues and delivery failures. If necessary, changes are managed systematically through Agile models.

2.6 Advantages and Disadvantages of SDLC

**2.6.1 Advantages of SDLC**

* **Structured Approach:** SDLC offers a structured and well-ordered framework for software development, ensuring that all stages are planned and executed.
* **Improved Quality Assurance:** Following predefined steps helps identify and resolve defects early, improving software quality.
* **Better Project Management:** SDLC facilitates better project management by breaking down each phase into clear milestones, deliverables, and timelines.
* **Enhanced Communication:** Stakeholders, including developers, testers, project managers, and customers, collaborate through SDLC, leading to better understanding of needs and requirements.
* **Risk Management:** Teams can use risk management techniques to identify potential risks and implement measures to minimize their impact.

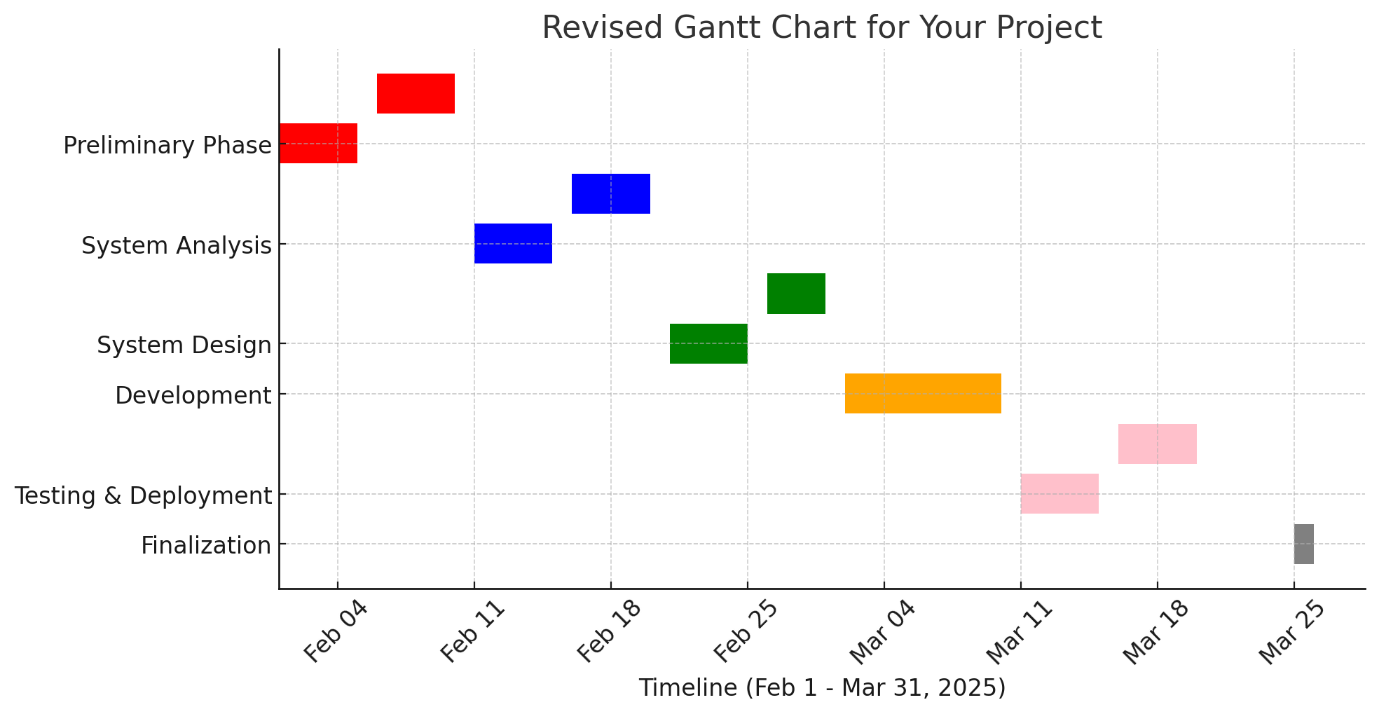
**2.6.2 Disadvantages of SDLC**

* **Rigidity:** SDLC may be rigid and inflexible, making it difficult to accommodate changes in requirements or adapt to shifting project needs, especially in fast-paced environments.
* **Time-Consuming:** The entire SDLC process, from requirements gathering to deployment, can be time-consuming.
* **High Cost:** Compared to Agile methods, SDLC often requires a high upfront investment in planning, documentation, and resources, leading to increased development costs.
* **Limited Stakeholder Involvement:** Sometimes, stakeholder involvement is limited to certain stages, leading to a lack of continuous feedback and collaboration.
* **Overemphasis on Documentation:** Excessive focus on documentation can divert resources from actual development activities.

**2.7 Waterfall: Linear & Sequential**

The Waterfall methodology proceeds through software delivery in orderly, consecutive phases without overlap. Stages include gathering requirements, system design, coding, testing, release, and maintenance. Its defined path brings discipline but restricts flexibility in shifting needs post-launch.

# **3.System Planning**



3.1 Preliminary Phase (Early February)

* **3.1.1 Project Search:** Identify a suitable project idea that aligns with your goals and requirements.
* **3.1.2 Requirement of Project:** Define the objectives, scope, and key functionalities needed in the project.

3.2 System Analysis (Mid-February)

* **3.2.1 Information Gathering:** Collect data from users, stakeholders, or reference materials to understand system requirements.
* **3.2.2 Data Modeling:** Structure the collected information using models like Entity-Relationship Diagrams (ERDs) or Data Flow Diagrams (DFDs).

3.3 System Design (Late February - Early March)

* **3.3.1 Design Building:** Create wireframes, prototypes, or sketches of the system's user interface and experience.
* **3.3.2 Charts & Diagram Preparation:** Develop detailed system architecture, flowcharts, and diagrams for database structures.

3.4 System Development (Mid-March)

* **3.4.1 System Coding:** Begin actual coding and implementation of the system’s core functionalities.

3.5 Testing & Implementation (Late March)

* **3.5.1 Testing:** Run unit, integration, and system tests to identify and fix bugs.
* **3.5.2 Implementation:** Deploy the system in a controlled environment or for real-world use.

3.6 Submission (End of March - Early April)

* **3.6.1 Final Review:** Conduct a final assessment of the system and refine any remaining issues.
* **3.6.2 Documentation:** Prepare comprehensive documentation for the project.
* **3.6.3 Submission:** Submit the completed project.

This structured approach ensures systematic progress through different phases while maintaining deadlines.

# **4.System Analysis**

4.1.1Project Overview  
The import-export project aims to create a seamless, efficient platform for businesses to manage international trade operations. The system will provide a user-friendly interface, real-time shipment tracking, WhatsApp API integration, and robust SEO optimization to ensure high visibility and smooth operations.

4.1.2 Functional Requirements

**User Interface & Experience (UI/UX)**

* Modern and visually appealing UI with engaging animations.
* Fully responsive design, ensuring seamless functionality on mobile and desktop devices.
* Easy-to-use navigation structure with clear CTA buttons.
* Dark mode and customizable themes for user preferences.
* Interactive dashboards for users to track shipments and manage orders.
* Multi-language support to cater to international users.

**SEO Optimization**

* On-page SEO best practices for better search rankings.
* Keyword-optimized content for import-export business visibility.
* Fast page loading speeds, reducing bounce rates and improving user retention.
* Structured data and schema markup for better indexing by search engines.
* Optimized URL structures and metadata to enhance discoverability.
* Automatic sitemap generation and XML submission to search engines.

**Shipment Tracking System**

* Real-time shipment tracking integrated with major logistics providers (DHL, FedEx, UPS, etc.).
* Live updates on shipment status: "Processing," "In Transit," "Customs Clearance," "Delivered."
* Tracking ID generation and lookup functionality.
* Customer notifications via WhatsApp, email, and SMS at key shipment milestones.
* Estimated delivery time calculations based on live tracking data.
* Historical shipment tracking for past orders.

**WhatsApp API Integration**

* Automated order confirmations and shipment status updates via WhatsApp.
* Two-way customer communication for queries and support.
* Integration with AI chatbots for automated responses and FAQ handling.
* Secure end-to-end encryption for all messages.
* Multi-user access for customer service teams to handle inquiries efficiently.
* Support for media-rich messages (images, PDFs, invoices, etc.).

**User Authentication & Management**

* Secure login with multi-factor authentication (MFA) support.
* Role-based access control (admin, business user, customer, logistics provider).
* OAuth and social login options (Google, Facebook, LinkedIn, etc.).
* User dashboard with shipment history, active orders, and notifications.
* Password recovery and account verification mechanisms.

**Admin Dashboard**

* Centralized dashboard for managing user accounts, shipments, and orders.
* Shipment analytics with performance insights.
* SEO performance tracking with keyword ranking reports.
* API integration management for third-party services.
* System logs and activity tracking for security monitoring.
* Bulk order and shipment data import/export functionality.

**Payment & Invoicing (Optional Future Scope)**

* Integration with international payment gateways (PayPal, Stripe, Razorpay, etc.).
* Auto-generated invoices with export compliance details.
* Tax and duty calculations based on shipment origin and destination.

4.1.3 Non-Functional Requirements

**Performance & Scalability**

* Page loading speed of under 2 seconds for all main operations.
* Scalable cloud infrastructure to handle increased traffic and transactions.
* Optimized animations to prevent UI lag.
* Support for high concurrent users without performance degradation.

**Security & Data Protection**

* End-to-end encryption for sensitive user and shipment data.
* Role-based access control (RBAC) for secure data handling.
* Secure API connections with token-based authentication.
* Compliance with GDPR, CCPA, and other international data protection laws.
* Regular security audits and vulnerability assessments.

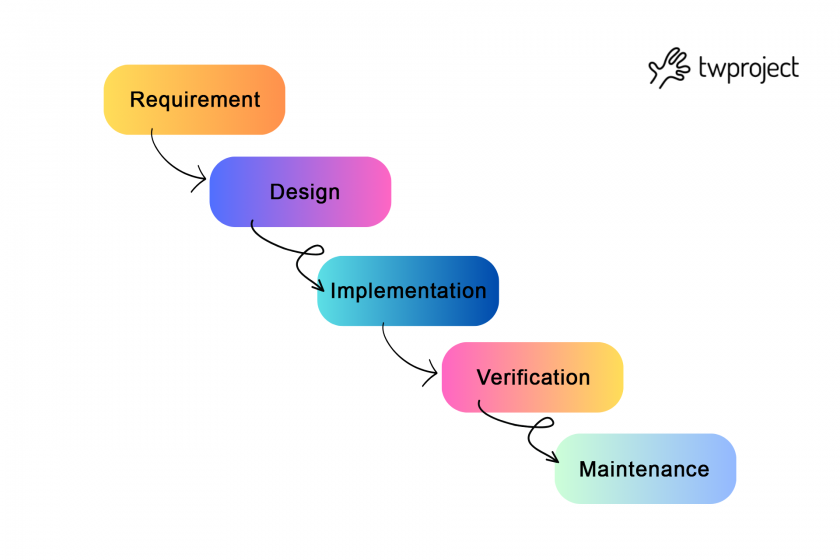
**Availability & Reliability**

* 99.9% uptime guarantee with redundant cloud-based hosting.
* Automated backups with quick recovery mechanisms.
* Load balancing to ensure uninterrupted service during peak hours.
* Disaster recovery plan for handling system failures.

**Maintainability & Extensibility**

* Modular, microservices-based architecture for easy updates and feature expansion.
* API-first approach allowing smooth third-party service integrations.
* Version-controlled deployment with minimal downtime.
* Regular updates based on user feedback and technological advancements.

4.2 Modal – Waterfall Model



**Requirement Gathering & Analysis**

* Gather detailed business and technical requirements.
* Define functional and non-functional requirements (UI/UX, SEO, shipment tracking, WhatsApp API integration, security, etc.).
* Identify third-party services (shipment tracking APIs, WhatsApp API providers).
* Prepare requirement specification documents.

**System Design**

* UI/UX wireframes and prototypes for web and mobile views.
* Database design for user accounts, shipments, and notifications.
* API integration design (WhatsApp API, shipment tracking API).
* Security framework setup (encryption, authentication methods).
* Defining server requirements (hosting, cloud infrastructure).
* SEO strategy planning.

**Implementation (Development)**

* Develop the frontend (React, Vue.js, or Angular) with animations.
* Build the backend (Node.js, Python, or PHP) to handle user authentication, order processing, and tracking.
* Implement database (MySQL, PostgreSQL, or MongoDB) for storing shipments, user data, and logs.
* Integrate WhatsApp API for automated messaging.
* Implement shipment tracking system using third-party APIs.
* Optimize SEO elements (meta tags, page speed improvements).
* Conduct unit testing on individual components.

**Testing**

* Functional testing – Ensure all features work as intended (shipment tracking, messaging, etc.).
* UI/UX testing – Validate animations, responsiveness, and ease of navigation.
* Performance testing – Check page load times, API response times, and database queries.
* Security testing – Ensure data encryption, role-based access, and API security.
* SEO testing – Verify if pages are indexed properly and optimized for search engines.
* Bug fixing – Address any issues found during testing.

**Deployment**

* Deploy the application on a cloud server or hosting platform.
* Configure domain, SSL certificates, and security settings.
* Monitor initial user feedback and fix any post-deployment issues.
* Ensure API integrations (shipment tracking, WhatsApp) work in real-time.
* Submit the site for SEO indexing (Google Search Console, etc.).

**Maintenance & Support**

* Monitor system uptime and performance.
* Release security patches and software updates.
* Enhance features based on user feedback.
* Optimize SEO performance regularly.
* Provide customer support for troubleshooting issues.

# 4.3 Feasibility study

### Feasibility Study for Import-Export Business Website

This document evaluates the feasibility of developing and deploying the Import-Export Business Website. It examines the project from technical, costing, and risk perspectives, ensuring that every aspect is within the available resources and capabilities.

### 1. Introduction

The Import-Export Business Website is designed to automate order processing, enable real-time shipment tracking, and integrate WhatsApp for instant customer support. With a modern UI/UX, an admin dashboard for insights, and SEO optimization, the project aims to streamline trade operations for businesses. This feasibility study demonstrates that the project is not only conceptually strong but also viable from technical, financial, and risk-management standpoints.

### 2. Technical Feasibility

**Definition**

**Technical Feasibility** assesses whether the technology, tools, and expertise required to develop and support the system are available and sufficient for the project’s scope.

**Evaluation**

* **Technology Stack:**
  + **Frontend:** The project uses HTML, CSS, JavaScript (with libraries such as GSAP/Three.js for animations) ensuring a modern, responsive, and interactive user interface.
  + **Backend:** Django and Django REST Framework are selected, providing a robust, secure, and scalable environment to handle business logic, data management, and API integration.
  + **Database:** PostgreSQL is used, which is well-suited for managing complex data requirements such as orders and shipment tracking.
  + **Additional Tools:** Integration of QR code/barcode libraries for tracking and the WhatsApp API for communication solidifies the real-time operational capacity.
* **Availability of Expertise:**  
  The required technical skills for web development, database management, and API integration are available. The chosen technologies are widely adopted, meaning documentation and community support are plentiful.
* **Infrastructure:**  
  Deployment on cloud services like DigitalOcean or AWS ensures high availability, scalability, and ease of maintenance.

### 3. Costing Feasibility

**Definition**

**Costing Feasibility** (or Economic Feasibility) examines whether the financial resources required for the project are justified by the anticipated benefits and whether the project fits within budget constraints.

**Evaluation**

* **Development Costs:**
  + **Software and Tools:** Most of the core technologies (Django, PostgreSQL, HTML/CSS/JavaScript libraries) are open-source or have low licensing fees.
  + **Infrastructure:** Cloud hosting costs (DigitalOcean/AWS) are scalable and can be optimized based on traffic and usage.
  + **Integration Expenses:** The cost for APIs (like WhatsApp) and third-party services are within typical budget ranges for small to medium business projects.
* **Operational Costs:**
  + Maintenance, regular updates, and security audits are factored into the ongoing costs. However, with the automation of many tasks (e.g., order processing, shipment tracking), operational costs are expected to remain manageable.
* **Return on Investment (ROI):**
  + Enhanced customer engagement and streamlined operations are likely to reduce manual errors and improve overall business performance.
  + The SEO-optimized platform is expected to drive more organic traffic, potentially increasing revenue.

### 4. Risk Assessment

**Definition**

**Risk Assessment** identifies potential challenges and uncertainties that could affect the project’s success. It also outlines strategies for mitigating these risks.

**Evaluation**

* **Technical Risks:**
  + **Integration Issues:**
    - *Risk:* Potential challenges in integrating third-party APIs (e.g., WhatsApp API, shipment tracking APIs).
    - *Mitigation:* Thorough testing during the development phase and leveraging well-documented APIs reduce this risk.
  + **Security Vulnerabilities:**
    - *Risk:* Data breaches or unauthorized access.
    - *Mitigation:* Implementing robust security measures (encryption, multi-factor authentication, regular security audits) ensures the system remains secure.
* **Cost-Related Risks:**
  + **Budget Overruns:**
    - *Risk:* Unexpected expenses in cloud hosting or additional tool licenses.
    - *Mitigation:* A phased development approach with regular budget reviews can control costs. Many core technologies are open-source, reducing potential overruns.
* **Operational Risks:**
  + **System Downtime:**
    - *Risk:* Unplanned downtime could affect business operations and user experience.
    - *Mitigation:* Use of reliable cloud services with 99.9% uptime, along with redundant systems and regular backups, minimizes downtime risks.
  + **User Adoption:**
    - *Risk:* Users may find it challenging to adapt to new system features.
    - *Mitigation:* A user-friendly design and detailed user guides, along with training sessions, can ease the transition.

## 4.4 Software Requirement Specification

Below is a detailed Software Requirements section for the Import-Export Business Website project. It builds upon the synopsis while adding more detail to ensure clarity and precision.

### 1. Introduction

The Import-Export Business Website is designed to automate order processing, enable real-time shipment tracking, and facilitate efficient communication through WhatsApp integration. The SRS provides a clear blueprint of the system’s functionalities and performance targets, serving as a guide for developers and stakeholders throughout the project lifecycle.

### 2. Functional Requirements

These requirements define what the system must do:

* **User Authentication & Management:**
  + Secure registration, login, password recovery, and multi-factor authentication.
  + Role-based access control for different user types (admin, business user, customer, logistics provider).
* **Dashboard & Admin Panel:**
  + An intuitive admin interface to monitor orders, track shipment statuses, and manage user accounts.
  + Real-time analytics and reporting features for informed decision-making.
* **Shipment Tracking:**
  + Real-time shipment updates using QR/barcode scanning and integration with tracking APIs.
  + Automated generation and lookup of tracking IDs.
* **Communication Module:**
  + Integration with the WhatsApp API for sending order confirmations and live shipment updates.
  + Support for two-way communication to handle customer inquiries.
* **Order Management:**
  + Automated processing of orders with status notifications and historical data logging.
  + Seamless coordination between order placement, shipment, and delivery tracking.
* **SEO & Content Management:**
  + Built-in SEO features such as optimized URL structures, metadata management, and automatic sitemap generation to enhance online visibility.

### 3. Non-Functional Requirements

These requirements specify the quality attributes the system must have:

* **Performance:**
  + Fast page loading times (target under 2 seconds for major operations) and smooth UI interactions across devices.
* **Scalability:**
  + A cloud-based infrastructure (DigitalOcean or AWS) capable of handling increased traffic and transactions as the business grows.
* **Security:**
  + Robust security measures including data encryption, secure API connections, regular security audits, and role-based access to protect sensitive information.
* **Availability:**
  + A high uptime guarantee (approximately 99.9%), along with backup and disaster recovery strategies to ensure continuous service.
* **Maintainability:**
  + A modular codebase with clear documentation to support easy updates, troubleshooting, and future enhancements.

### 4. Constraints & Dependencies

* **Third-Party Integrations:**
  + Reliance on external services such as the WhatsApp API, QR/barcode libraries, and cloud hosting providers requires thorough testing and periodic updates.
* **Resource Allocation:**
  + The system must be developed within defined budgetary and time constraints, aligning with the planned phased approach for testing and deployment.

## 4.5 Tools and Technologies

**Software Specification**

This section outlines the key technologies employed in the project. Each technology has been chosen to ensure the website is modern, scalable, secure, and user-friendly.

**Frontend: HTML, CSS, and JavaScript**

HTML structures the content, CSS styles the visual elements, and JavaScript brings interactivity. Libraries like GSAP and Three.js are used to create smooth animations and dynamic effects, ensuring a visually engaging and responsive user interface.

**WhatsApp API**

The WhatsApp API enables direct, automated messaging with customers. It is used for sending order confirmations and real-time shipment updates, ensuring quick and efficient communication that enhances overall customer support.

**Backend: Django & Django REST Framework**

Django serves as the core backend framework, offering robust security and rapid development using the Model-View-Template pattern. The Django REST Framework extends its capabilities by simplifying API creation, which is essential for real-time tracking and seamless integration with external services.

**Database: PostgreSQL**

PostgreSQL is chosen as the database management system for its reliability, performance, and advanced features. It efficiently handles complex transactional data such as orders and shipment tracking details, ensuring data integrity and smooth operations.

**Tracking System: QR/Barcode Libraries & Scanners**

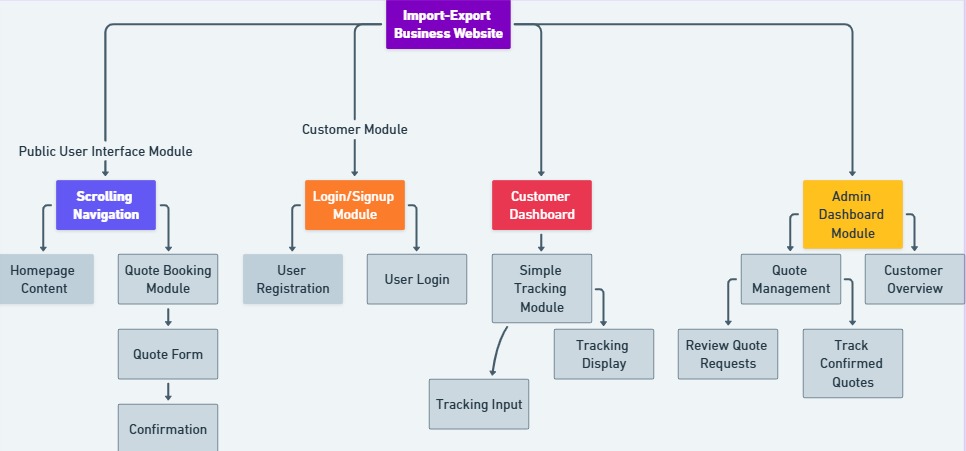
QR and barcode libraries (e.g., qrcode, pyzbar) are used to generate and scan tracking IDs, facilitating real-time shipment tracking. Integrated RFID and barcode scanners further automate status updates, improving accuracy and efficiency in logistics management.

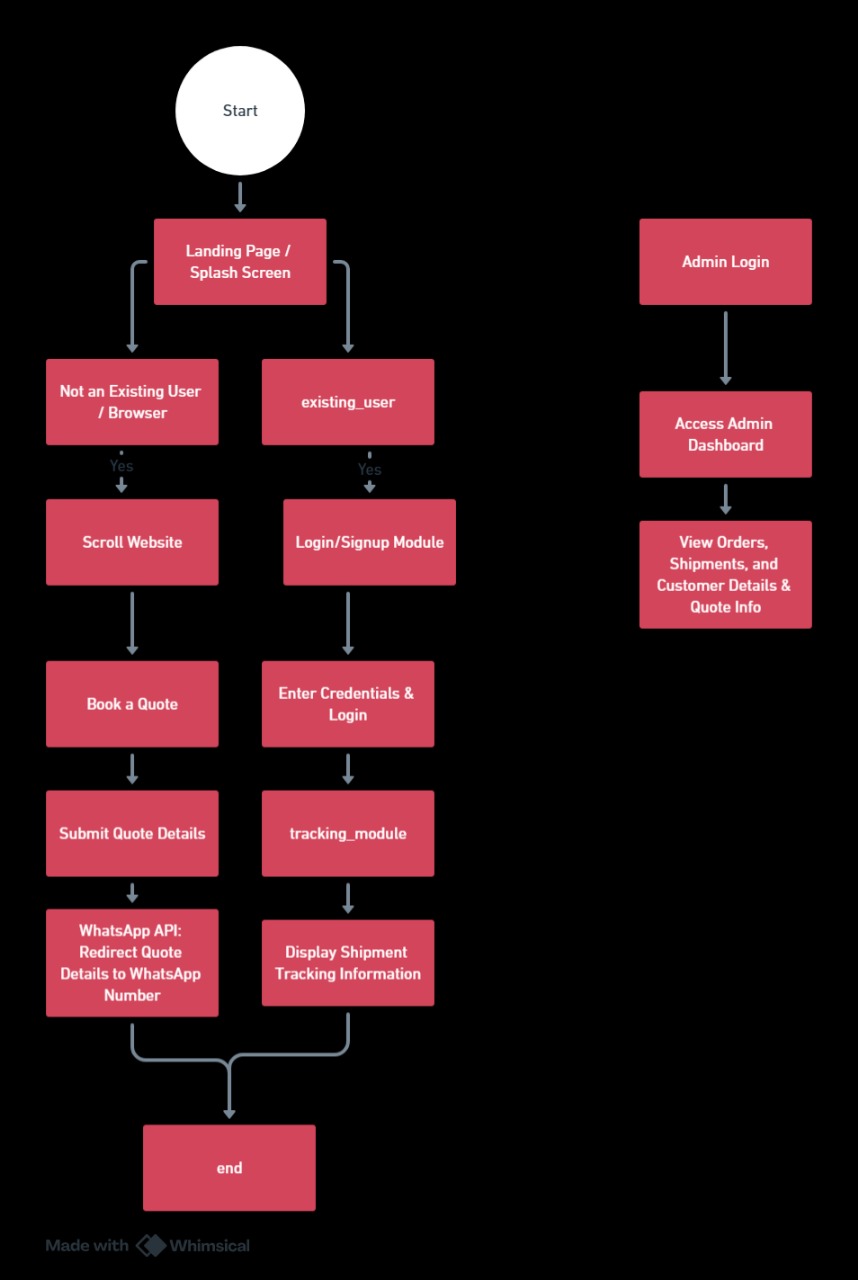
**Deployment: DigitalOcean/AWS, Gunicorn, and Nginx**

The project is deployed on cloud platforms like DigitalOcean or AWS to ensure scalability and high availability. Gunicorn acts as the Python WSGI HTTP server, while Nginx serves as a reverse proxy and load balancer, creating a robust, high-performance production environment.

# **5. System Design**

## 5.1 Module decomposition tree:



5.2 Flow chart

## 5.4 Data Modelling:

### ER Diagram

**Entities and Relationships**

* **User**
  + Attributes: user\_id, username, email, password, registration\_date
  + **Relationship:** One user can have many quotes and many orders.
* **Quote**
  + Attributes: quote\_id, user\_id, quote\_details, quote\_date, accepted\_status
  + **Relationship:** Each quote is submitted by one user (one-to-many: User → Quote).
* **Order**
  + Attributes: order\_id, user\_id, order\_date, order\_status
  + **Relationship:** One user can have many orders.
* **Shipment**
  + Attributes: shipment\_id, order\_id, tracking\_number, shipment\_date, shipment\_status
  + **Relationship:** Each order is associated with one shipment (one-to-one or one-to-many if needed, but here we assume one order results in one shipment).

## 5.5 Database Schema: Data Table Headers

**User Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| user\_id | INT / SERIAL | PRIMARY KEY, Auto Increment |
| username | VARCHAR(50) | NOT NULL, UNIQUE |
| email | VARCHAR(100) | NOT NULL, UNIQUE |
| password | VARCHAR(255) | NOT NULL |
| registration\_date | DATE/TIMESTAMP | DEFAULT current\_timestamp |

**Quote Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| quote\_id | INT / SERIAL | PRIMARY KEY, Auto Increment |
| user\_id | INT | FOREIGN KEY (references User.user\_id) |
| quote\_details | TEXT | NOT NULL |
| quote\_date | DATE/TIMESTAMP | DEFAULT current\_timestamp |
| accepted\_status | BOOLEAN or ENUM | NOT NULL (e.g., 'Pending', 'Accepted', 'Rejected') |

**Order Table**

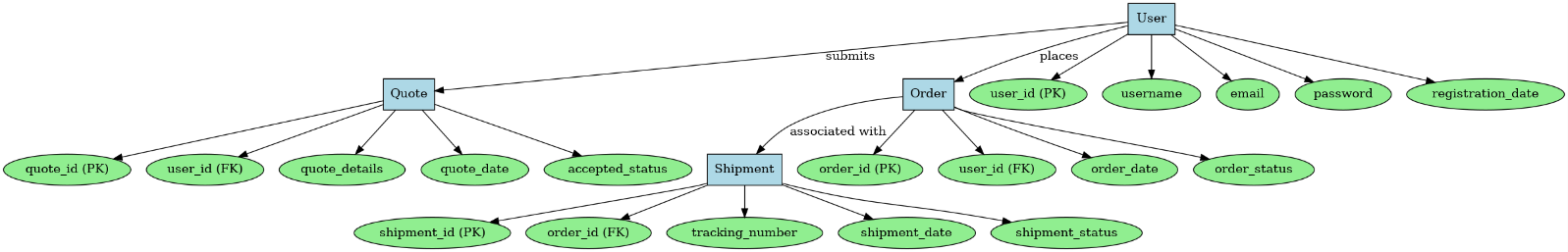
|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| order\_id | INT / SERIAL | PRIMARY KEY, Auto Increment |
| user\_id | INT | FOREIGN KEY (references User.user\_id) |
| order\_date | DATE/TIMESTAMP | DEFAULT current\_timestamp |
| order\_status | VARCHAR(50) or ENUM | NOT NULL (e.g., 'Pending', 'Confirmed') |

**Shipment Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| shipment\_id | INT / SERIAL | PRIMARY KEY, Auto Increment |
| order\_id | INT | FOREIGN KEY (references Order.order\_id) |
| tracking\_number | VARCHAR(50) | NOT NULL, UNIQUE |
| shipment\_date | DATE/TIMESTAMP | DEFAULT current\_timestamp |
| shipment\_status | VARCHAR(50) or ENUM | NOT NULL (e.g., 'In Transit', 'Delivered', etc.) |

**Conventions and Relationship Summary**

* **One-to-Many:**
  + **User → Quote:** One user can submit multiple quotes.
  + **User → Order:** One user can place multiple orders.
* **One-to-One (or One-to-Many if extended):**
  + **Order → Shipment:** Each order is associated with one shipment.



# **8.Implementation**

# **8.1.User Guide**

### **Introduction**

GlobalLog is a web application designed for managing import/export logistics. This user guide provides detailed instructions on how to navigate and use the website effectively.

### **1. User Registration & Login**

**1.1 Sign Up**

1. Navigate to the **Sign Up** page.
2. Enter the following details:
   * Username
   * Email
   * Password
3. Click **Register**.
4. After successful registration, you can log in using your credentials.

**1.2 Login**

1. Navigate to the **Login** page.
2. Enter your **Username** and **Password**.
3. Click **Login**.
4. If credentials are correct, you will be redirected to the dashboard.
5. If incorrect, an error message will appear.

### **2. Dashboard**

* After logging in, you will land on the **Dashboard**.
* The dashboard provides access to various features:
  + Quote Requests
  + Orders
  + Shipments
  + User Profile

### **3. Quote Request**

**3.1 Submitting a Quote Request**

1. Navigate to the **Quote Request** section.
2. Enter the required details:
   * Material Type
   * Tonnage
   * Container Size
   * Container Type
   * Location
3. Click **Submit**.
4. The request will be saved in the database and sent via WhatsApp to the logistics team.

**3.2 Viewing Quote Status**

1. Navigate to **My Quotes**.
2. Check the status of your quote (Pending, Accepted, or Rejected).

### **4. Shipments**

**4.1 Viewing Shipment Details**

1. Navigate to **Shipments**.
2. Track shipment status (In Transit, Delivered, etc.).
3. View tracking number and shipment location.

### **5. User Profile & Settings**

**5.1 Updating Profile**

1. Click on **Profile** in the navigation bar.
2. Update your details (Username, Email, Password).
3. Click **Save Changes**.

**5.2 Logging Out**

1. Click on **Logout** in the navigation bar.
2. You will be redirected to the login page.

### **6. Error Handling & Troubleshooting**

* **Invalid Login Credentials**: Ensure you are using the correct username and password.
* **Quote Not Submitting**: Check if all required fields are filled.
* **Order Not Placing**: Ensure the quote has been approved.
* **Shipment Not Found**: Verify if the order has been confirmed.

### **7. Contact Support**

**For further assistance, contact customer support via email at support@globallog.com.**