Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Title of the Project

Development of an E-Post Office System

Abstract of the project

The rapid advancements in digital technology have significantly transformed service-based industries, making it essential for traditional services such as postal systems to modernize and adapt to the changing landscape. This project, titled **E-Post Office**, aims to design and implement an efficient, web-based postal service management system, leveraging modern web development technologies including **React.js** for the front-end interface, **Node.js** for the back-end server environment, and **MongoDB** for database management.

The core objective of the **E-Post Office** is to provide a seamless online platform where users can access essential postal services. Key features of the system include **real-time package tracking**, **online postage purchasing**, **scheduling parcel pick-ups**, and comprehensive **user account management**. The application focuses on eliminating the challenges associated with physical postal services by offering a faster, more convenient, and user-friendly alternative.

The use of React.js ensures a highly dynamic, interactive, and responsive user interface, while Node.js powers the server-side operations, managing user requests, and interacting with MongoDB to store and retrieve data. The overall system architecture ensures scalability, security, and ease of use, delivering an enhanced digital postal experience to users.

Keywords

Generic Keywords

Databases, Front-End, Back-End

Specific Technology Keywords

React.js, Node.js, HTML, CSS, JavaScript

Project Type keywords

Analysis, Design, Implementation, Testing, Graphical User Interface

Functional components of the project

The **E-Post Office Web Application** consists of several key functional components designed to provide users with a wide range of postal services, while ensuring system

security, efficiency, and ease of use. Below is a detailed list of the system's core functionalities:

1. Users of the System:

- **Customer (End-User)**: The primary user of the system, customers can log in to the platform, manage their accounts, track packages, and schedule postal services such as pick-ups and deliveries.

2. Welcome Page:

- When users enter the URL of the website, they are greeted with a **Welcome Page**.
- The page includes the following elements:
  - **Menu**: Located on the left-hand side, it provides easy navigation to various sections such as Home, Services, My Account, Package Tracking, etc.
  - **Banner**: Displayed at the top, showcasing important notices, promotions, or service updates.
  - **Related Links**: A section containing links to relevant external websites or other online services offered by the postal system.

3. User Authentication (Login and Registration):

- **Login Functionality**: Users must log in to the system using valid credentials (email and password) before accessing any services.
  - The system checks the **authenticity of the user** by comparing the provided credentials against the information stored in the **MongoDB** database.
  - A failed login attempt prompts the user to retry or recover their password through a secure password reset mechanism.
- **User Registration**: New users must register by providing personal details such as name, address, email, and password. Once registered, they can access all available postal services after verification.

4. User Dashboard (Post-Login):

- After successful login, the user is redirected to their **Dashboard** where they can:
  - **View Profile**: Manage personal information such as name, address, and contact details.
  - **View Order History**: Review past postal transactions, including package details, payment history, and delivery status.
  - **Real-Time Package Tracking**: Enter tracking numbers and get real-time updates on the location and status of parcels.

5. Package Tracking:

- The system integrates with external **logistics APIs** to provide real-time tracking of packages.
  - Users can view current package location, estimated delivery time, and any relevant status updates.
  - Admins can update package statuses, adding entries such as "In Transit," "Out for Delivery," or "Delivered."

6. Online Postage Purchasing:

- Users can calculate postage based on package dimensions and weight and pay for postage services online.
    - **Payment Gateway Integration**: Secure online payments through **payment gateways** (such as **Stripe** or **PayPal**) allow users to complete their transactions and print shipping labels directly from the platform.

7. Contact Us Functionality:

- The **Contact Us** page provides a form where users can submit queries or issues regarding postal services.
    - The form allows users to enter their **name, email, subject**, and **message**.
    - On submission, the form data is stored in the MongoDB database, and an admin can review and respond to the queries.
    - Users receive an acknowledgment email once their query is submitted, and an admin can track and reply to these inquiries through the admin panel.
    - This functionality ensures effective communication between users and the postal service team, improving user support and satisfaction.

Following is a list of functionalities of the system. More functionality that you find appropriate can be added to this list. And, in places where the description of functionality is not adequate, you can make appropriate assumptions and proceed.

Menu Structure and Detailed Screen Descriptions

When the user types the URL of the website, they are greeted with a **Welcome Page**, featuring the following key elements:

- **Menu**: Displayed on the left-hand side of the screen, this menu provides users easy navigation through various sections of the system.
- **Banner**: Located at the top of the page, featuring important notices, promotional messages, or system updates.
- **Related Links**: Links to external resources or relevant postal services.

Users must log in from the Welcome Page before accessing any services or placing orders. The **Login Functionality** ensures that user credentials are authenticated by verifying them against the information stored in the **MongoDB** database. If users are not registered, they will be redirected to the **Registration Screen**.

1. Registration Screen

If a user is not registered, the system will display a **Registration Screen** that allows them to create an account. Users are required to input personal information such as their name, address, email, and password. Once the registration is successful, the user is redirected to the **Home** page. The information provided during registration is stored in the **MongoDB** database for future authentication.

2. Complaint

This screen provides a form for users to submit complaints related to the services. Users can fill out details regarding their issue, including selecting the type of service (e.g., package delivery, postage issues) and submitting relevant order or tracking numbers. Complaints are recorded in the database for administrators to review and respond to in a timely manner.

3. Contact Us
The **Contact Us** page includes a form for users to send queries or feedback. The form captures basic details like **name, email, subject**, and **message**, and upon submission, the query is stored in the database. Administrators can manage and respond to queries through the admin panel. Users will also receive an acknowledgment email confirming that their message was received.

4. Home
The **Home** page serves as the user's main dashboard after logging in. From here, users can access essential features such as **Tracking Orders**, viewing the **Catalogue**, managing their account, and more. It provides a summary of their current orders, postal activity, and recent transactions.

5. Products
The **Products** page allows users to explore available postal products and services, such as postage options, stamp orders, and bulk shipment packages. Users can browse, select, and add items to their cart, where pricing and quantity will be dynamically managed.

6. Cart
The **Cart** displays all the products selected by the user, including stamps, letters, and other postal items. It dynamically updates the total price based on selected quantities and items. Users can review their selected items before proceeding to the checkout process. Once confirmed, payment is processed securely through the integrated payment gateway.

7. Quick Links
This section offers users shortcuts to important sections of the system, such as **Help**, **Frequently Asked Questions (FAQ)**, **Terms & Conditions**, and other external postal-related services.

Steps to start-off the project

Technology Stack:

- **Front-end**: React.js (JavaScript framework for building user interfaces)
- **Back-end**: Node.js with Express (server-side and API)
- **Database**: MongoDB (data management)

Steps to Initiate:

1. **Familiarize with the Technology Stack**: Gain a solid understanding of **React.js**, **Node.js with Express**, and **MongoDB** for development.
    - **Resources**:
        - React.js Documentation
        - Node.js & Express Documentation
        - MongoDB Documentation

2. **Define User Profiles and Business Rules**:
    - Identify user types: regular customers and administrators.
    - Specify product types (stamps, letters) and set operational rules for orders, registration, and complaints.

3. **Ensure User-Friendly Help**:
    - Design intuitive help sections that guide users through each feature.

4. **Maintain UI Consistency**:
    - Use a consistent design with clear images, intuitive navigation, and uniform styling across all screens.

Requirements

Hardware requirements

Number	Description	Alternatives (If available)
1	PC with 256 GB hard-disk and 16 GB RAM	Not-Applicable

Software requirements

Number	Description	Alternatives (If available)
1	Windows 11	Mac OS
2	MongoDB	Not Applicable
3	Node.js	Not Applicable
4	Visual Studio Code (IDE for development)	WebStorm, Sublime Text

Manpower requirements

- **Team Size**: 2 to 3 students
- **Project Duration**: 4 – 6 months if worked on full-time

Milestones and Timelines

- **React.js Documentation**: https://reactjs.org/docs/getting-started.html
- **Node.js Documentation**: https://nodejs.org/en/docs/
- **MongoDB Documentation**: https://www.mongodb.com/docs/