**MEDLEY PHARMA**

**INTRODUCTION**

In a world driven by the synergy of technology and convenience, where every aspect of our lives seamlessly converges within the digital realm, the field of healthcare is swiftly evolving to meet the ever-growing needs of patients around the globe. Allow us to introduce the groundbreaking "Medley Pharma" initiative – a pioneering leap towards enhancing pharmaceutical accessibility, efficacy, and patient-centered service.

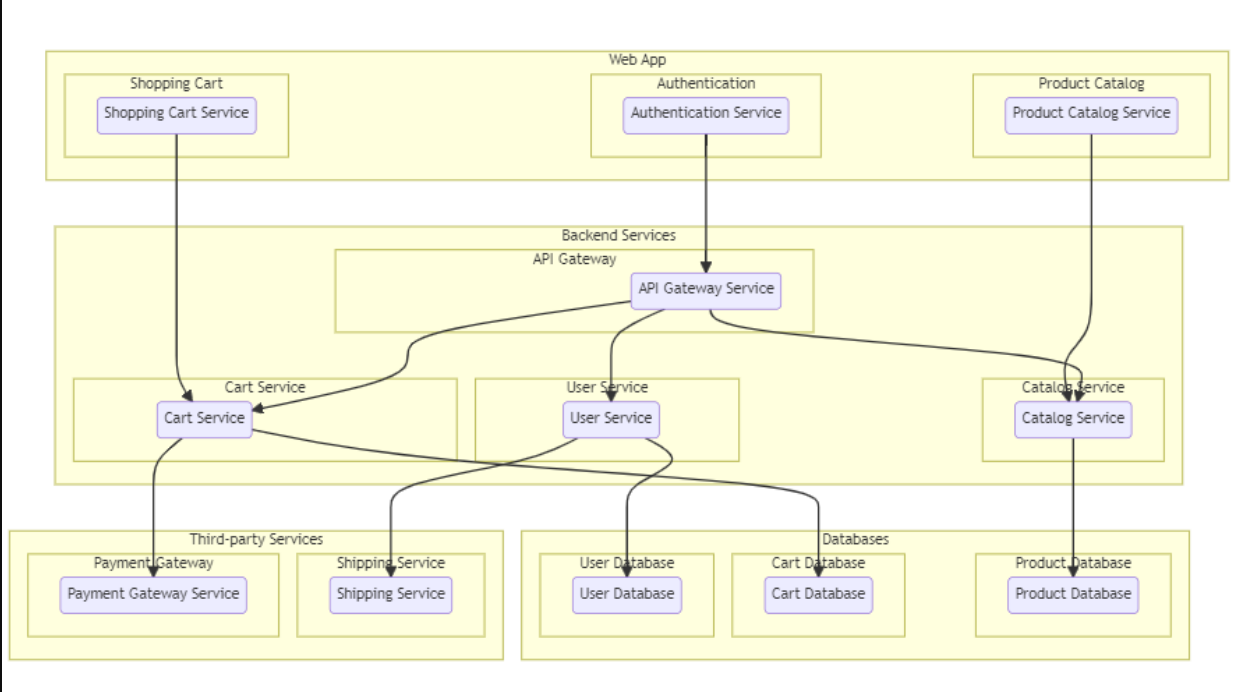
Say goodbye to the era of lengthy waits, endless phone calls, and frustrating prescription processes. Our state-of-the-art platform adeptly harnesses the power of modern technology to seamlessly connect patients with their prescribed medications within a framework that is sleek, intuitive, and user-friendly. Whether you require regular prescription refills, specialized treatments, or prompt access to essential medications, the "Medley Pharma" project emerges as the guiding light set to revolutionize your experience with pharmaceuticals.

At the heart of the "Medley Pharma" initiative lies a vision of a pharmaceutical ecosystem that deeply values each patient's invaluable time and holistic well-being. By bridging the gap between patients and essential medications, our mission is to curate a pharmaceutical journey marked by convenience, efficiency, and an unwavering focus on patient welfare. Our platform goes beyond mere prescription management; it aspires to empower patients to take control of their health and wellness journey.

As we embark on this transformative journey into the future of pharmaceutical accessibility, we extend a sincere invitation to you to collaborate with us in creating a realm where access to vital medications is as effortless as a few clicks on your preferred device. The "Medley Pharma" initiative transcends being a mere digital interface; it embodies our resolute commitment to redefining pharmaceutical accessibility on a global scale. Rediscover pharmaceuticals from a fresh perspective. Your well-being, your schedule – all guided by your choices.

The "Medley Pharma" initiative represents a monumental stride in the realm of pharmaceutical accessibility. Through the seamless integration of cutting-edge frontend and backend technologies, it transforms the patient experience, streamlining the process of securing prescribed medications with just a few clicks. This innovative approach to connecting patients with medications holds the latent power to reshape the very fabric of pharmaceutical services, potentially setting a new benchmark for transformation across diverse sectors.

**TECHNICAL ARCHITECTURE**



## ER-DIAGRAM

The Entity-Relationship (ER) diagram for "Medley Pharma" visually illustrates the fundamental data entities—patients, doctors, appointments—along with their relationships and attributes. This diagram provides a concise overview of how these entities interact within the platform's data structure, enhancing understanding and design clarity.

1. Identify Entities

2. Define Relationships

3. Draw Diagram

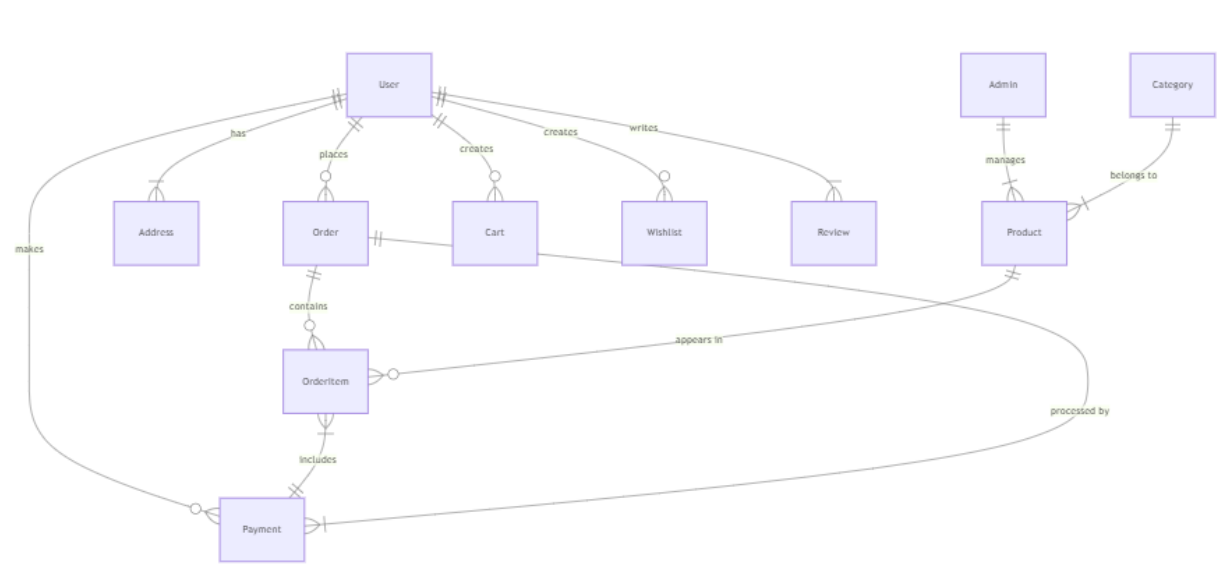
4. Label and Connect

5. Refine and Review

6. Share and Discuss

7. Document and Implement

ER Diagram:



**Entity description:**

1. User:

- User\_ID (Primary Key)

- Name

- Contact Information

- Address

2. Medicine:

- Medicine\_ID (Primary Key)

- Name

- Description

- Dosage Instructions

- Price

3. Order:

- Order\_ID (Primary Key)

- User\_ID (Foreign Key referencing User)

- Order Date

- Total Amount

4. Order Item:

- Item\_ID (Primary Key)

- Order\_ID (Foreign Key referencing Order)

- Medicine\_ID (Foreign Key referencing Medicine)

- Quantity

- Subtotal

**Relationships:**

* A User can place zero or more Orders. (User to Order, One-to-Many)
* An Order belongs to one User. (Order to User, Many-to-One)
* An Order consists of one or more Order Items. (Order to Order Item, One-to-Many)
* An Order Item corresponds to one Medicine. (Order Item to Medicine, Many-to-One)
* You can represent these entities and relationships using an ER diagram drawing tool like Lucidchart, draw.io, or any other similar software. In the diagram, entities are usually represented as rectangles, attributes as ovals, and relationships as lines connecting the entities with appropriate annotations to indicate the nature of the relationship (1:1, 1:N, N:M).
* Remember, the above outline is a basic representation. Depending on the complexity of your system, you might need to add more entities and attributes. You could also include additional details like user authentication, payment processing, and more. The key is to design the ER diagram in a way that accurately represents the structure of your medicines ordering system.

**Key features:**

1. **User Authentication and Profiles:-** Users, which can include patients and healthcare providers, can create and manage profiles. Profiles contain personal information and medication history.

2. **Effortless Medicine Ordering:-** Users can easily browse and select from a variety of medicines available on the platform, add them to their cart, and proceed to place orders.

3. **Comprehensive Medicine Details**: - Each medicine listing includes detailed information such as the medicine's name, dosage instructions, possible side effects, and user reviews.

4. **Flexible Delivery Options**:- Users can choose between different delivery methods, including standard shipping and expedited delivery, to receive their medications at their preferred location.

5. **Medication History Management**:- Users can securely store and access their medication history, including past orders, dosage information, and any potential interactions with other medications.

6. **Transparent Pricing**:- The platform displays the cost of each medicine clearly, allowing users to make informed decisions based on their budget.

7. **Medicine Ordering**: - Design an interface that enables users to search for medicines, add them to their cart, provide prescription details if required, and confirm the order.

8. **User Feedback**:- Allow users to provide feedback and reviews for medicines they've ordered, helping other users make informed choices.

**PRE REQUISITES**

To develop a full-stack medley pharma app using AngularJS, Node.js, and MongoDB, there are several prerequisites you should consider. Here are the key prerequisites for developing such an application:

**Node.js and npm:**Install Node.js, which includes npm (Node Package Manager), on your development machine. Node.js is required to run JavaScript on the server side.

* Download:<https://nodejs.org/en/download/>
* Installation instructions:<https://nodejs.org/en/download/package-manager/>

**MongoDB**:Set up a MongoDB database to store hotel and booking information. Install MongoDB locally or use a cloud-based MongoDB service.

* Download:<https://www.mongodb.com/try/download/community>
* Installation instructions:<https://docs.mongodb.com/manual/installation/>

**Express.js**:Express.js is a web application framework for Node.js. Install Express.js to handle server-side routing, middleware, and API development.

* Installation: Open your command prompt or terminal and run the following command:npm install express

**Angular**:Angular is a JavaScript framework for building client-side applications. Install Angular CLI (Command Line Interface) globally to create and manage your Angular project.

 Install Angular CLI:

* Angular provides a command-line interface (CLI) tool that helps with project setup and development.
* Install the Angular CLI globally by running the following command:

npm install -g @angular/cli

**Verify the Angular CLI installation**:

* Run the following command to verify that the Angular CLI is installed correctly: ng version

You should see the version of the Angular CLI printed in the terminal if the installation was . successful.

**Create a new Angular project**:

* Choose or create a directory where you want to set up your Angular project.
* Open your terminal or command prompt.
* Navigate to the selected directory using the cd command.
* Create a new Angular project by running the following command: ng new client Wait for the project to be created: The Angular CLI will generate the basic project structure and install the necessary dependencies.

**Navigate into the project directory:**

* After the project creation is complete, navigate into the project directory by running the following command: cd client .

**Start the development server**:

* To launch the development server and see your Angular app in the browser, run the following command:ng serve / npm start.
* The Angular CLI will compile your app and start the development server.
* Open your web browser and navigate to http://localhost:4200 to see your Angular apprunning.

You have successfully set up Angular on your machine and created a new An- gular project. You can now start building your app by modifying the generated project files in the src directory.

Please note that these instructions provide a basic setup for Angular. You can explore more ad- vanced configurations and features by referring to the official Angular documentation: <https://angular.io>

 HTML, CSS, and JavaScript:Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

 Database Connectivity:Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.

 Front-end Framework:Utilize Angular to build the user-facing part of the application, including products listings, booking forms, and user interfaces for the admin dashboard.

Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

* Git:Download and installation instructions can be found at:<https://git-scm.com/downloads>

Development Environment:Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

* Visual Studio Code: Download from[https://code.visualstudio.com/download](http://./%20https:/code.visualstudio.com/download%20)
* Sublime Text: Download from<https://www.sublimetext.com/download>
* WebStorm: Download from<https://www.jetbrains.com/webstorm/download>

**To Connect the Database with Node JS go through the below provided link**:

 • Link:[https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/](https://www.section.io/engineering-education/nodejs-%20mongoosejs-mongodb/)

## To run the existing Expense Tracker App project downloaded from github:

**Follow below steps:**

## Clone the Repository:

* + Open your terminal or command prompt.
  + Navigate to the directory where you want to store the medicine booking app.
  + Execute the following command to clone the repository:

**git clone** : <https://github.com/charani197/Medley-Farma-Application>

## Install Dependencies:

* + Navigate into the cloned repository directory:
  + cd medley pharma
  + Install the required dependencies by running the following command:

npm install

## Start the Development Server:

* + To start the development server, execute the following command:

npm run dev or npm run start

* + The e-commerce app will be accessible at http://localhost:5100 by default. You can change the port configuration in the .env file if needed.

## Access the App:

* + Open your web browser and navigate to http://localhost:5100.
  + You should see the medicine booking app's homepage, indicating that the installation and setup were successful.

**Video Tutorial Link to clone the project: -**

**Project Repository Link:**  <https://github.com/charani197/Medley-Farma-Application>

Congratulations on successfully installing and setting up the medicine booking app on your local machine, marking the start of an exciting journey in app development. With the app now operational, you can proceed to customize and refine it to suit your specific requirements. This is an opportunity to enhance the user interface, expand features, and conduct rigorous testing to ensure a seamless user experience. By iteratively improving and testing the app, you can create a polished and user-friendly product that exceeds expectations.

As you venture into further customization, development, and testing, remember that each enhancement contributes to the app's overall quality and appeal. Fine-tune the user interface, add features, and thoroughly test for any issues. Your dedication to this process will result in an app that not only meets but surpasses user needs, setting the stage for a successful and engaging medicine booking experience.

**Roles and Responsibilities:**

**Admin Role**:

* Responsibilities: The admin role has full control and administrative privileges over the system.
* Permissions:
  + Manage: Admins can add, edit, and delete shop information along with products.
  + Manage product bookings: Admins can view and manage all product bookings made by users and agents, including canceling or modifying product bookings.
  + Manage users: Admins can create, edit, and delete user accounts, as well as manage their roles and permissions.
  + Generate reports: Admins have access to generate reports and analytics on product booking details, booking counts, and sales reports.

**User Role**:

* Responsibilities:Users are the customers of the online shopping web application who can search for products, and make product bookings.
* Permissions:
  + View products: Users can search for products, based on interest.
  + Product bookings: Users can select products that are available and complete the order process.
  + Manage product booking: Users can view their own product order bookings, modify booking details, track booking details, and cancel their bookings
  + Manage cart: Users can view their cart details and modify them if needed.

Roles and responsibilities can vary based on the specific project requirements and the level of administrative control desired. It's important to clearly define these roles to ensure smooth functioning of the "Medley Pharma" platform while maintaining a positive user experience.

**USER FLOW**

**User Flow: Ordering Medicines**

**User Registration and Login**:

1. User visits the platform's website.

2. If new, user registers with their details (name, email, password).

3. If already registered, user logs in with their credentials.

**Browse Medicines**:

4. User is redirected to the dashboard or medicine listings page.

5. User can see a list of medicines with their names, dosages, and prices.

6. User can click on a medicine to view more details.

**View Medicine Details:**

7. User sees medicine details: name, dosage instructions, possible side effects, and user reviews.

8. User can read other user reviews and ratings for the medicine.

**Select Medicines:**

9. User adds the desired medicines to their cart.

10. The platform displays the selected medicines in the user's cart.

**Review Cart and Proceed:**

11. User reviews the cart to ensure the selected medicines are correct.

12. User proceeds to the checkout process.

**Provide Prescription (if required):**

13. If certain medicines require a prescription, user provides the necessary prescription details.

14. If not required, user proceeds without providing a prescription.

**Confirm Order:**

15. User confirms the order details, including medicines, quantities, and prescription information.

16. User clicks "Place Order" to proceed.

**Order Confirmation and Notification:**

17. User receives a confirmation message with order details.

18. User may receive an email or SMS notification as well.

**View and Manage Orders**:

19. User can view all their placed orders in their profile.

20. User can see order details, including medicines, quantities, and order status.

**Edit or Cancel Order:**

21. If needed, user can edit or cancel orders within a defined time frame.

22. User receives a confirmation for any changes made.

**Write Review:**

23. After receiving and using the medicines, user has the option to write a review for each medicine.

24. User rates the medicine and adds comments if desired.

**User Profile:**

25. User can access their profile to update personal information.

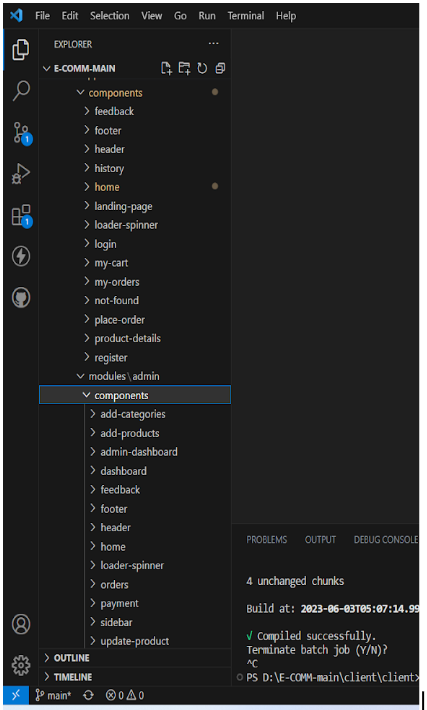
26. User can view their order history, reviews, and preferences.

**Logout:**

27. User can log out when done using the platform.

This adapted user flow outlines the steps a user would take to order medicines through the medicine booking system. It covers the process of browsing medicines, adding them to the cart, providing any required prescription information, confirming the order, managing orders, and interacting with the user profile. Just like before, you can customize and expand this flow based on the specific features and functionalities of your medicine booking system.

**PROJECT STRUCTURE**:



## The project structure may vary depending on the specific framework, programming language, or development approach used. It's essential to organize the files and directories in a logical and consistent manner to improve code maintainability and collaboration among developers.

app/app.component.scss, src/app/app.component.spec.ts: These files are part of the main AppComponent, which serves as the root component for the Angular app. The component handles the overall layout and includes the router outlet for loading different

components based on the current route.

## PROJECT FLOW:

**Milestone 1: Project Setup and Configuration:**

### Install required tools and software:

* + Node.js.
  + MongoDB.
  + Angular CLI.

### Create project folders and files:

* + Client folders.
  + Server folders.

## Milestone 2: Backend Development:

### Setup express server:

* + Install express.
  + Create app.js file.
  + Define API’s

### Configure MongoDB:

* + Install Mongoose.
  + Create database connection.
  + Create Models.

### Implement API end points:

* + Implement CRUD operations.
  + Test API endpoints.

## Milestone 3: Web Development:

### Setup Angular Application:

* + Create Angular application using angular CLI
  + Configure Routing.
  + Install required libraries.

### Design UI components:

* + Create Components.
  + Implement layout and styling.
  + Add navigation.

### Implement frontend logic:

* + Integration with API endpoints.
  + Implement data binding.

## BACKEND

**Set Up Project Structure:**

* Create a new directory for your project and set up a package.json file using npm init command.
* Install necessary dependencies such as Express.js, Mongoose, and other required packages.

**Database Configuration:**

* Set up a MongoDB database either locally or using a cloud-based MongoDB service like MongoDB Atlas.
* Create a database and define the necessary collections for hotels, users, bookings, and other relevant data.

**Create Express.js Server:**

* Set up an Express.js server to handle HTTP requests and serve API endpoints.
* Configure middleware such as body-parser for parsing request bodies and cors for handling cross-origin requests.

**Define API Routes:**

* Create separate route files for different API functionalities such as hotels, users, bookings, and authentication.
* Define the necessary routes for listing hotels, handling user registration and login, managing bookings, etc.
* Implement route handlers using Express.js to handle requests and interact with the database.

**Implement Data Models:**

* Define Mongoose schemas for the different data entities like hotels, users, and bookings.
* Create corresponding Mongoose models to interact with the MongoDB database.
* Implement CRUD operations (Create, Read, Update, Delete) for each model to perform database operations.

**API Design and Development:**

* Identify the necessary functionality and data required by the frontend.
* Design a set of RESTful APIs using a framework like Express.js or Django REST Framework.
* Define API endpoints for user management, product catalog, shopping cart, order management, payment gateway integration, shipping integration, etc.
* Implement the API routes, controllers, and data models to handle the corresponding operations.
* Ensure that the APIs follow best practices, are secure, and provide appropriate responses.

**User Management and Authentication:**

* Implement user registration and login functionality.
* Choose an authentication mechanism like session-based authentication or token-based authentication (e.g., JWT).
* Store and hash user credentials securely.
* Implement middleware to authenticate API requests and authorize access to protected routes.

**Product Catalog and Inventory Management:**

* Design the database schema to store product details, pricing, availability, and inventory levels.
* Create APIs to retrieve product information, update inventory quantities, and handle search and filtering.
* Implement validations to ensure data integrity and consistency.

**Shopping Cart and Order Management:**

* Design the database schema to store shopping cart details and order information.
* Create APIs to handle cart operations like adding items, modifying quantities, and placing orders.
* Implement logic to calculate totals, apply discounts, and manage the order lifecycle.

**Payment Gateway Integration:**

* Choose a suitable payment gateway provider (e.g., Stripe, COD).
* Integrate the payment gateway SDK or API to handle secure payment processing.
* Implement APIs or callback endpoints to initiate transactions, handle payment callbacks, and receive payment confirmation.

**Shipping and Logistics Integration:**

* Identify shipping and logistics providers that align with your application's requirements.
* Utilize the APIs provided by these providers to calculate shipping costs, generate shipping labels, and track shipments.
* Implement APIs or services to fetch rates, generate labels, and obtain tracking information.

**Database Integration:**

* Choose a suitable database technology (e.g., MySQL, PostgreSQL, MongoDB) based on your application's requirements.
* Design the database schema to efficiently store and retrieve flower and gift delivery data.
* Establish a connection to the database and handle data persistence and retrieval.

**External Service Integration:**

* Identify third-party services like email service providers, analytics services, or CRM systems that are required for your application.
* Utilize the APIs or SDKs provided by these services to exchange data and perform necessary operations.
* Implement the integration logic to send order confirmations, track user behavior, or manage customer relationships.

**Security and Data Protection:**

* Apply appropriate security measures like encryption techniques for secure data transmission and storage.
* Implement input validation and sanitization to prevent common security vulnerabilities.
* Implement access control to ensure authorized access to sensitive data.

**Error Handling and Logging:**

* Implement error handling mechanisms to handle exceptions and provide meaningful error messages to the frontend.
* Use logging frameworks to record application logs for monitoring and troubleshooting purposes.

**KEY FEATURES**

**User Authentication and Profiles**: At the heart of the "Order Medicines" platform lies a robust user authentication and profile management system. Patients and healthcare providers can create and manage personalized profiles, enabling tailored experiences. This empowers users to take ownership of their healthcare journey and ensures secure access to the platform's offerings.

**Comprehensive Medicine Listings**: A treasure trove of medical solutions, the platform's expansive database showcases a wide range of medicines. This comprehensive repository includes vital information about each medicine, including its name, dosage instructions, potential side effects, and user reviews. These reviews act as guiding lights for users as they navigate the landscape of medication options.

**Smart Medicine Selection**: The backend's intelligence shines through its adept utilization of advanced algorithms. These algorithms analyze user preferences and medication availability to suggest optimal medicine options. By considering various factors, they help users make well-informed choices quickly, ensuring efficient and convenient medicine selection.

**Real-time Notifications**: The platform thrives on real-time notifications that keep users informed. These automated messengers ensure that users are up to date with their order status, shipment tracking, and other essential updates. Whether it's order confirmations, changes, or delivery information, these notifications act as vigilant guides, providing timely information.

**Secure Data Handling**: Security is paramount in the healthcare realm, and the "Order Medicines" platform prioritizes it. Stringent measures protect user information, enveloping it in layers of encryption and security protocols. Adherence to regulatory standards guarantees the integrity and confidentiality of user data.

**User-Friendly Interface**: The frontend interface embodies a design philosophy that emphasizes user-friendliness and aesthetics. Navigational pathways are intuitively designed to ensure easy access for users of varying technical backgrounds. A visually appealing design isn't just about aesthetics—it's a commitment to user-centricity, ensuring seamless navigation through the platform.

As the "Order Medicines" project unfolds, these intricately woven functionalities create a dynamic platform for medication empowerment. User profiles, medicine databases, smart selection algorithms, real-time notifications, security protocols, and an intuitive frontend come together to transcend ordinary medicine ordering. This symphony of technology and healthcare orchestrates a harmonious ecosystem where each element resonates with the shared ideals of convenience, accessibility, and excellence.

**CONCLUSION**

The Medley Pharma's advanced medicines booking system stands as a testament to innovation, illuminating the healthcare landscape with its transformative brilliance. A guiding light for accessible healthcare, it signifies the evolution from traditional medicine procurement methods to a digital realm where convenience and efficiency unite in perfect harmony. At its core, this project exemplifies the seamless integration of cutting-edge frontend and backend technologies, birthing a healthcare experience that redefines the ordinary.

Central to this project is the recognition of the dynamic fusion between technology and healthcare, a convergence that has the potential to redefine how patients engage with the world of medicine. The days of complex and time-consuming medicine ordering are now a relic of the past, replaced by a simple action – a mere tap or click. The elegant amalgamation of contemporary frontend design and robust backend architecture has crafted an environment where the user's journey is not only frictionless but also empowering.

This system transcends mere convenience; it acts as a bridge connecting patients with a network of medicines and healthcare professionals, fostering meaningful relationships. By offering patients the autonomy to choose medicines based on detailed information, including dosage, effects, and user reviews, the platform empowers well-informed decisions. In parallel, healthcare providers gain an effective tool to manage inventory and engage with patients, nurturing relationships that extend beyond a transactional level.

However, Medley Pharma's medicines booking system is more than just a localized advancement. Its impact stretches across industries, serving as a model for others to follow. The fusion of technology and convenience provides a template that goes beyond healthcare, demonstrating how diverse sectors can uplift user experiences through digital innovations.

As healthcare takes a leap forward, the system acts as a catalyst for broader discussions on the potential for transformation across multiple domains.

In summary, Medley Pharma's medicines booking system showcases the power of innovation in reshaping healthcare dynamics. It champions accessibility, efficiency, and patient-centricity not merely as ideals but as the foundations of a new era in medicine procurement. As the system embarks on its journey, it leaves an indelible mark, inviting patients, providers, and industries to participate in an experience where technology and care intertwine, ushering in a brighter, more accessible future for all.