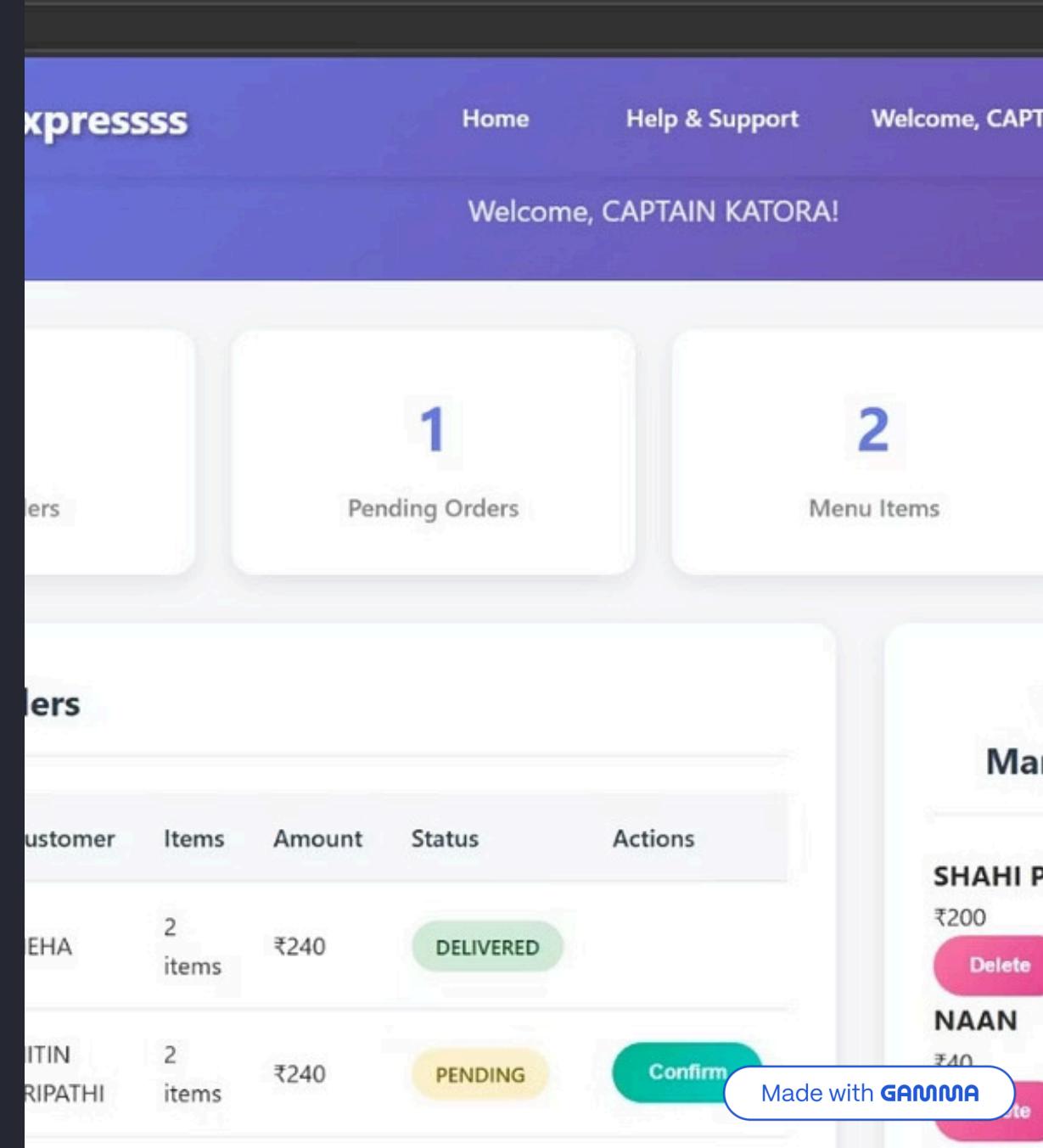


# NSS Food Delivery: A Prototype Showcase

Revolutionizing the way you experience food delivery.



# Our Vision for Seamless Food Experiences

## Enhanced User Experience

Creating an intuitive and delightful journey from browsing to delivery.

## Efficient Order Management

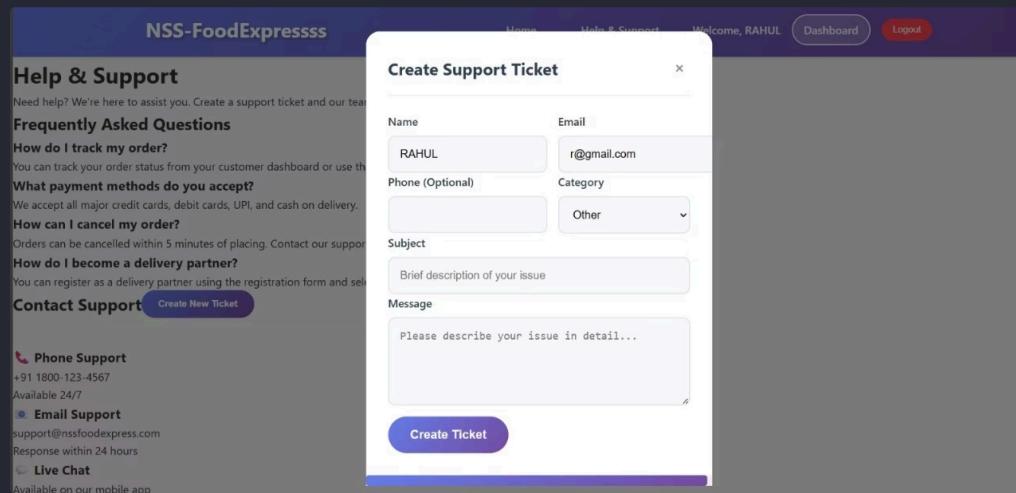
Streamlining processes for quick and accurate order fulfillment.

## Robust Platform

Building a scalable and reliable system for future growth.

This prototype is designed to address key challenges in food delivery, offering a glimpse into the future of convenience and culinary satisfaction.

# User-Centric Design: Homepage & Navigation



The homepage prioritizes **easy access to popular categories and personalized recommendations**.

- Clear search bar for quick restaurant or cuisine lookup.
- Prominent display of daily specials and featured partners.
- Intuitive navigation for seamless browsing.

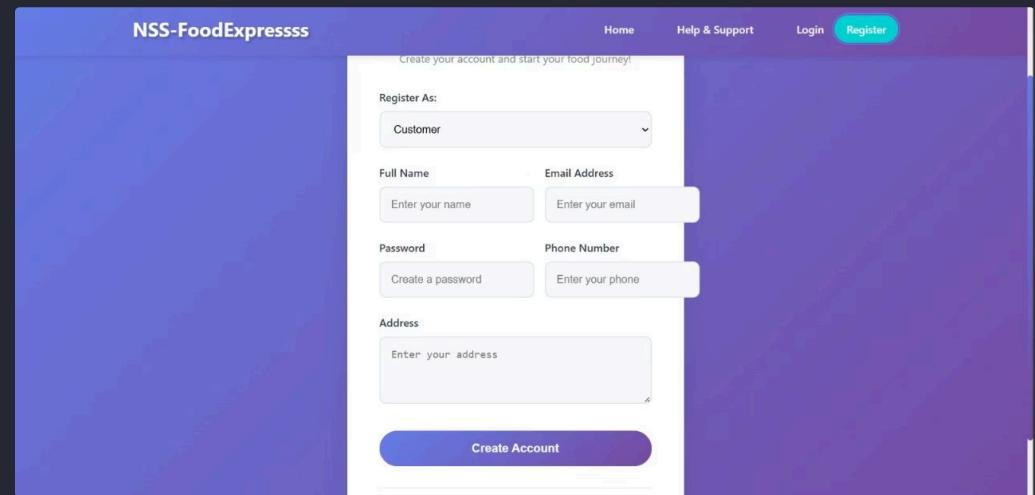
We aim to make finding your next meal effortless and enjoyable.

# Browsing Made Easy: Restaurant & Menu Views

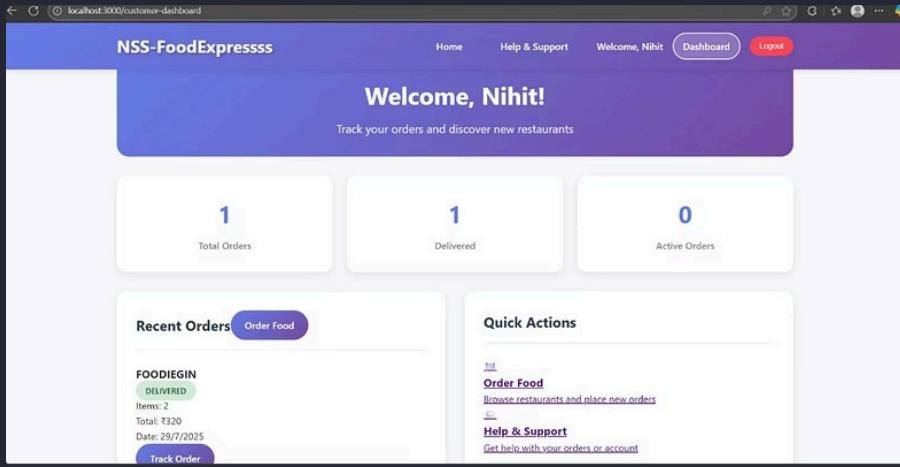
Our restaurant and menu pages are designed for clarity and comprehensive information.

- High-quality images of dishes to entice users.
- Detailed descriptions, ingredients, and allergen information.
- Filter and sort options for dietary preferences and price.

This ensures users make informed decisions with confidence.



# Effortless Ordering: Cart & Checkout



The screenshot shows the customer dashboard interface. At the top, there's a header with the logo 'NSS-FoodExpressss' and navigation links: Home, Help & Support, Welcome, Nihit, Dashboard, and Logout. The main area has a purple header 'Welcome, Nihit!' with the sub-instruction 'Track your orders and discover new restaurants'. Below this are three summary cards: 'Total Orders' (1), 'Delivered' (1), and 'Active Orders' (0). A 'Recent Orders' section shows a single order for 'FOODIEGIN' labeled 'DELIVERED' with details: Items: 2, Total: \$320, Date: 29/7/2025, and a 'Track Order' button. To the right is a 'Quick Actions' sidebar with 'Order Food' (link to 'Browse restaurants and place new orders'), 'Help & Support' (link to 'Get help with your orders or account'), and a 'Logout' button.

```
frontend > src > pages > CustomerDashboard.js > ...  
1 import React, { useState, useEffect } from 'react';  
2 import { Link } from 'react-router-dom';  
3 import axios from 'axios';  
4 import { useAuth } from '../context/AuthContext';  
5  
6 const CustomerDashboard = () => {  
7   const { user } = useAuth();  
8   const [orders, setOrders] = useState([]);  
9   const [loading, setLoading] = useState(true);  
10  
11   useEffect(() => {  
12     fetchOrders();  
13   }, [ ]);  
14  
15   const fetchOrders = async () => {  
16     try {  
17       const response = await axios.get('/orders/my');  
18       setOrders(response.data);  
19     } catch (error) {  
20       console.error('Error fetching orders:', error);  
21     } finally {  
22       setLoading(false);  
23     }  
24   };  
25 };
```

The cart and checkout flow prioritize **simplicity** and **security**.

- Real-time order summary and customizable options.
- Multiple payment methods, including digital wallets.
- Clear delivery time estimates and address management.

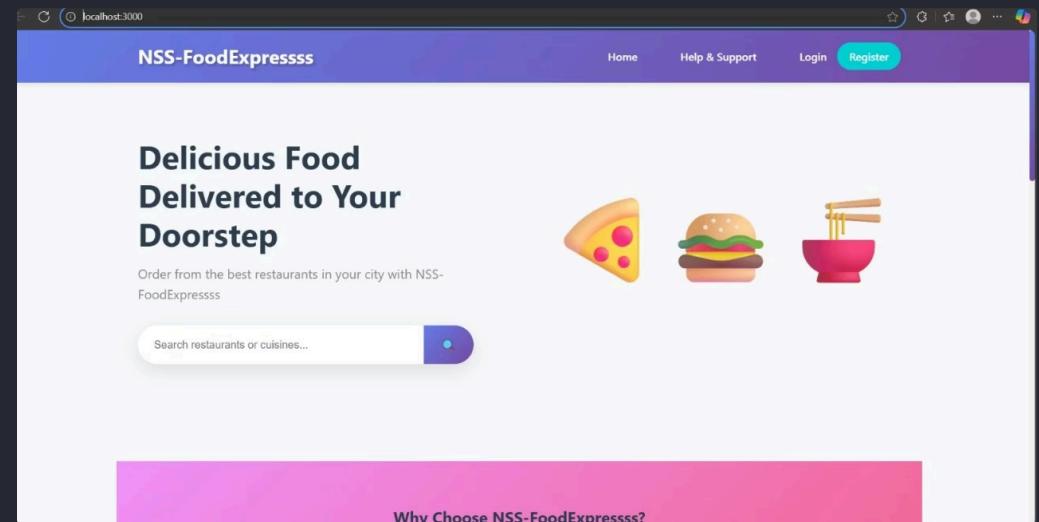
A smooth transaction process enhances user trust and satisfaction.

# Tracking Your Order: Real-time Updates

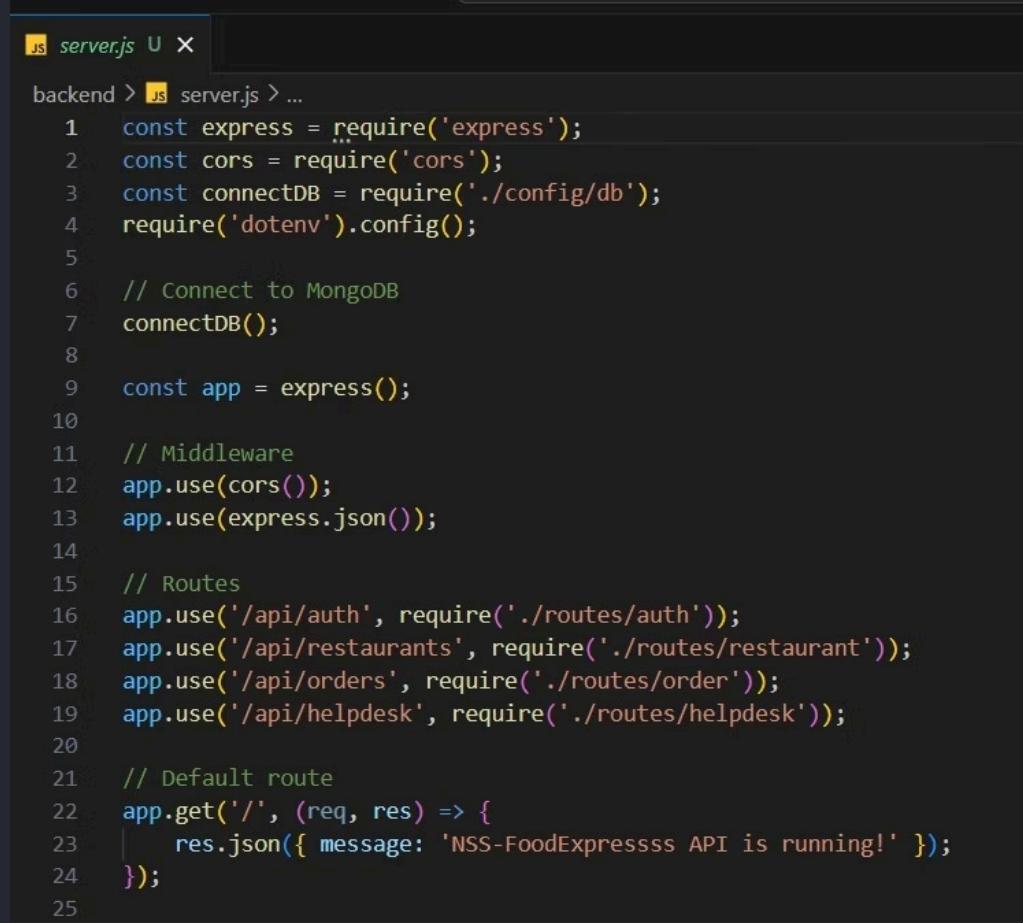
Stay informed every step of the way with our **real-time order tracking**.

- Live map view of your delivery driver's progress.
- Notifications for order preparation, dispatch, and arrival.
- Direct communication with the driver for any specific instructions.

Transparency builds confidence and reduces anxiety for users.



# Personalized Experience: User Profile & History



A screenshot of a code editor showing the `server.js` file. The code is a Node.js application using Express.js to handle requests. It includes middleware for CORS and JSON parsing, and routes for authentication, restaurants, orders, and helpdesk. A default route returns a JSON message indicating the API is running.

```
js server.js U X
backend > js server.js > ...
1 const express = require('express');
2 const cors = require('cors');
3 const connectDB = require('./config/db');
4 require('dotenv').config();
5
6 // Connect to MongoDB
7 connectDB();
8
9 const app = express();
10
11 // Middleware
12 app.use(cors());
13 app.use(express.json());
14
15 // Routes
16 app.use('/api/auth', require('./routes/auth'));
17 app.use('/api/restaurants', require('./routes/restaurant'));
18 app.use('/api/orders', require('./routes/order'));
19 app.use('/api/helpdesk', require('./routes/helpdesk'));
20
21 // Default route
22 app.get('/', (req, res) => {
23   res.json({ message: 'NSS-FoodExpressss API is running!' });
24 });
25
```

The user profile is a hub for personalized preferences and convenience.

- Quick re-ordering from past purchases.
- Saved addresses and payment methods.
- Loyalty points and special offers tracking.

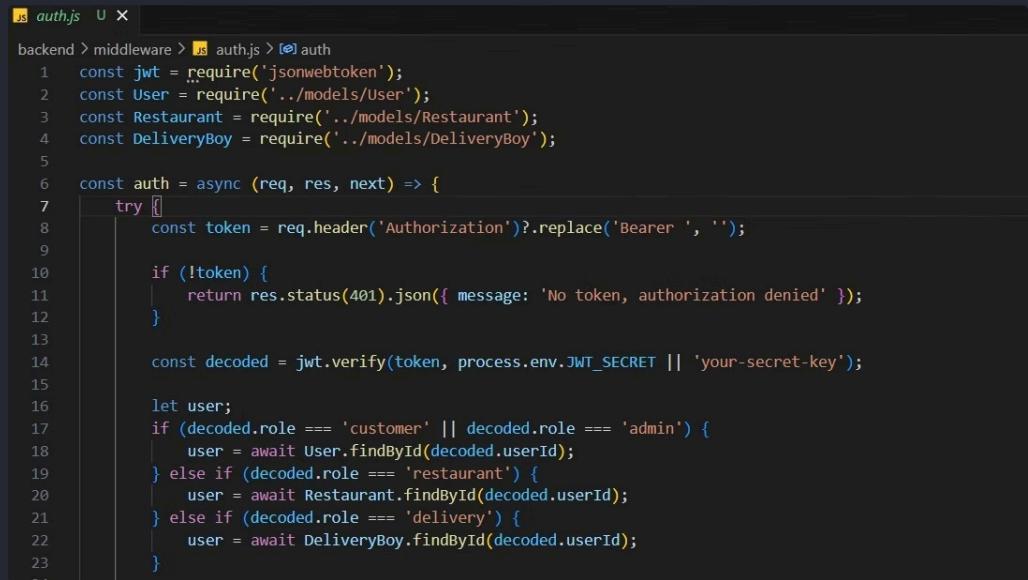
This feature fosters repeat business and user loyalty.

# Feedback & Support: Ensuring Satisfaction

We believe in continuous improvement through robust feedback mechanisms.

- Easy-to-use rating and review system for restaurants and drivers.
- In-app support chat for immediate assistance.
- FAQ section to answer common queries.

Your feedback helps us refine and enhance the NSS Food Delivery experience.



A screenshot of a code editor showing the `auth.js` file. The file is located in the `backend/middleware` directory. It contains code for handling JWT authentication. The code includes imports for `jsonwebtoken`, `User`, `Restaurant`, and `DeliveryBoy` models. It defines an asynchronous middleware function that checks for a token in the Authorization header, verifies it using a secret key, and then finds the user based on their role (customer, admin, restaurant, or delivery). The code uses promises and conditional logic to handle different user types.

```
js auth.js ✘ x
backend > middleware > js auth.js > auth
1 const jwt = require('jsonwebtoken');
2 const User = require('../models/User');
3 const Restaurant = require('../models/Restaurant');
4 const DeliveryBoy = require('../models/DeliveryBoy');
5
6 const auth = async (req, res, next) => {
7   try {
8     const token = req.header('Authorization')?.replace('Bearer ', '');
9
10    if (!token) {
11      return res.status(401).json({ message: 'No token, authorization denied' });
12    }
13
14    const decoded = jwt.verify(token, process.env.JWT_SECRET || 'your-secret-key');
15
16    let user;
17    if (decoded.role === 'customer' || decoded.role === 'admin') {
18      user = await User.findById(decoded.userId);
19    } else if (decoded.role === 'restaurant') {
20      user = await Restaurant.findById(decoded.userId);
21    } else if (decoded.role === 'delivery') {
22      user = await DeliveryBoy.findById(decoded.userId);
23    }
24  }
25}
```

# Key Innovations & Future Enhancements



## AI-Powered Recommendations

Personalized suggestions based on past orders and browsing.



## Drone Delivery Integration

Exploring faster delivery options for select areas.



## Sustainable Packaging Focus

Partnerships with eco-friendly suppliers for green deliveries.

We're committed to pushing boundaries and integrating cutting-edge technology to serve our users better.

# Next Steps & Call to Action

## Prototype Testing

Schedule a hands-on session to experience the prototype.

## Feedback Collection

Provide your valuable insights to shape future development.

## Strategic Planning

Collaborate on the roadmap for full platform development.

Let's build the future of food delivery together. Contact us to get involved!