Restora Digital Web App

Step-by-Step Project Documentation

Table of Contents

Introduction

This document provides a complete step-by-step guide to set up and use the Restora Restaurant Management System. Restora is a web-based application that helps restaurants manage their menu, orders, reservations, and customer reviews.

This guide uses simple language and clear instructions, making it easy for anyone to follow along, even if you are new to web development.

What is Restora?

Restora is a restaurant management system designed for French fine dining restaurants. It allows restaurant owners (admins) and customers (users) to interact through a web application.

For Restaurant Owners (Admins):

- Manage menu items (add, edit, delete)
- View and manage table reservations
- Track sales and orders
- View customer reviews

For Customers (Users):

- Browse the menu
- Add items to cart and place orders
- Make table reservations
- Leave reviews for menu items
- View order history

Prerequisites

Before you start, make sure you have the following software installed on your computer:

Required Software

1. **Node.js** (Version 18 or higher)

    - Download from: https://nodejs.org/
    - This allows you to run JavaScript on your computer
    - After installation, verify by opening a terminal/command prompt and typing: `node --version`
    - You should see a version number like "v18.x.x" or higher

2. **npm** (Node Package Manager)

    - This comes automatically with Node.js
    - Verify by typing: `npm --version`
    - You should see a version number

3. **A Code Editor** (Optional but recommended)

    - Visual Studio Code: https://code.visualstudio.com/
    - Or any text editor you prefer

4. **A Web Browser**

    - Chrome, Firefox, Edge, or Safari
    - Any modern browser will work

Optional Software

- **Git** (if you want to use version control)
- **Vercel Account** (if you want to deploy online)

Step-by-Step Setup Instructions

Step 1: Download or Clone the Project

1. If you have the project files in a ZIP folder:

    - Extract the ZIP file to a folder on your computer
    - For example: `C:\Users\YourName\Documents\Restora-main`

2. If you have the project from GitHub:

    - Open a terminal/command prompt
    - Navigate to where you want the project: `cd Documents`
    - Clone the repository: `git clone [repository-url]`
    - Navigate into the project: `cd Restora-main`

Step 2: Open the Project Folder

1. Open your terminal/command prompt

2. Navigate to the project folder:

```
cd path/to/Restora-main
```

Replace `path/to/Restora-main` with the actual path to your project folder.

Example on Windows:

```
cd C:\Users\YourName\Documents\Restora-main
```

Example on Mac/Linux:

```
cd /Users/YourName/Documents/Restora-main
```

Step 3: Check Node.js Installation

1. In the terminal, type:

```
node --version
```

2. You should see a version number. If you see an error, install Node.js first (see Prerequisites).

Step 4: Install Dependencies

1. The project uses Node.js but doesn't require external packages (dependencies are minimal)
2. However, if there's a `package.json` file, you can run:

```
npm install
```

This command reads the `package.json` file and installs any required packages.

Step 5: Start the Server

1. In the terminal, make sure you are in the project folder

2. Start the server by typing:

```
node server.js
```

Or if you have npm scripts set up:

```
npm start
```

3. You should see a message like:

```
Server running on http://localhost:3000
API endpoints available at /api/*
```

Step 6: Open the Application in Your Browser

1. Open your web browser (Chrome, Firefox, etc.)
2. Type in the address bar:

```
http://localhost:3000
```

3. Press Enter
4. You should see the Restora login page

Step 7: Test the Application

1. **Create a User Account:**

   - Click on "Sign Up" tab
   - Enter your name, email, and password
   - Click "Create My Account"
   - You should see a success message

2. **Login as Admin:**

   - Click on "Login" tab
   - Enter email: `admin` or `admin@gmail.com`
   - Enter password: `admin123456`
   - Click "Continue"
   - You should be redirected to the main page

3. **Login as Regular User:**

   - Use the email and password you created in Step 7.1
   - Click "Continue"
   - You should see the main page

Project Structure

Here's what each folder and file does:

```
Restora-main/
├── api/                      # Backend API files
│   ├── db.js                 # Database operations
│   ├── menu.js               # Menu management API
│   ├── orders.js             # Order management API
│   ├── reservations.js       # Reservation management API
│   ├── reviews.js            # Review management API
│   ├── sales.js              # Sales tracking API
│   ├── users.js              # User management API
│   └── notifications.js      # Email notifications API
│
├── assests/                   # Frontend files (note: "assests" is a typo,
should be "assets")
│       ├── images/            # Food images
│       └── main page/         # Main application pages
│           ├── index.html     # Main dashboard
│           ├── app.js         # Frontend JavaScript
│           ├── styles.css     # Styling
│           └── about.html     # About page
│
├── index.html                # Login page
├── server.js                 # Main server file
├── package.json              # Project configuration
└── vercel.json               # Deployment configuration
```

Important Files Explained

- **server.js**: This is the main server file. It handles all HTTP requests and serves the web pages.
- **api/**: This folder contains all the backend API endpoints. Each file handles a specific feature (menu, orders, etc.).
- **index.html**: This is the login page where users and admins sign in.
- **assests/main page/index.html**: This is the main dashboard after login.

Key Features

Restora has the following main features:

1. User Authentication

- **Sign Up**: New users can create accounts
- **Login**: Users and admins can log in
- **Admin Access**: Special admin account with extra permissions
- **Password Security**: Password strength checking

2. Menu Management

- **View Menu**: Browse all menu items

/

- **Search Menu**: Search for specific dishes
- **Filter by Category**: Filter menu by categories (Main Course, Desserts, etc.)
- **Admin Features**: Add, edit, and delete menu items (admin only)

3. Order Management

- **Shopping Cart**: Add items to cart
- **Place Orders**: Complete orders with payment
- **Order History**: View past orders
- **Bill Calculation**: Automatic tax calculation (5%)

4. Table Reservations

- **Make Reservations**: Book a table for a specific date and time
- **View Reservations**: See all your reservations
- **Admin Management**: Admins can approve or reject reservations

5. Reviews System

- **Leave Reviews**: Rate and review menu items (1-5 stars)
- **View Reviews**: See what others have said about dishes
- **Review Management**: Admins can delete reviews if needed

6. Sales Dashboard (Admin Only)

- **View Sales**: See total sales and revenue
- **Sales by Item**: See which items sell the most
- **Sales by Date**: Track daily sales

7. Notifications

- **Email Notifications**: Get emails for order confirmations, reservation updates, etc.
- **Note**: Currently uses mock email system (prints to console)

8. User Profile

- **View Profile**: See your account information
- **Update Profile**: Change your name and preferences

How to Use Each Feature

Using the Login Page

1. **Open the application** in your browser: `http://localhost:3000`
2. **To create an account:**
   - Click "Sign Up" tab
   - Enter your name, email, and password
   - Password must be at least 8 characters
   - Click "Create My Account"

3. **To login:**
    - Click "Login" tab
    - Enter your email and password
    - Click "Continue"

Using Menu Features

1. **View Menu:**

    - After login, you'll see the menu on the main page
    - Menu items show name, price, and image

2. **Search Menu:**

    - Type in the search box at the top
    - Results filter automatically

3. **Filter by Category:**

    - Use the dropdown menu to select a category
    - Only items from that category will show

4. **Add Items to Cart:**

    - Click the "+" button on any menu item
    - The item is added to your cart
    - You can adjust quantity in the cart

5. **Admin: Manage Menu:**

    - Login as admin
    - Click "Manage Menu" button
    - Add new items: Fill in name, price, image URL, category
    - Edit items: Click edit, change details, save
    - Delete items: Click delete button

Using Order Features

1. **Add to Cart:**

    - Click "+" on menu items
    - Items appear in the "Cart & Bill" section on the right

2. **View Cart:**

    - See all items in your cart
    - See subtotal, tax (5%), and total

3. **Place Order:**

    - Click "Pay Now" button
    - Select payment method (Cash, Card, UPI)

- Click "Confirm Payment"
- Order is placed and saved

4. **View Order History:**

- Click "Order History" button
- See all your past orders
- See order details, status, and total

Using Reservation Features

1. **Make a Reservation:**

- Click "Reserve Table" button
- Fill in the form:
  - Your name
  - Phone number
  - Number of guests
  - Date (pick a date)
  - Time (pick a time)
  - Special requests (optional)
- Click "Book Table"
- Reservation is created

2. **View Your Reservations:**

- Click "View Reservations" button (if you have reservations)
- See all your bookings
- See status: pending, approved, or rejected

3. **Admin: Manage Reservations:**

- Login as admin
- View all reservations from all users
- Approve or reject reservations
- Change reservation status

Using Review Features

1. **Leave a Review:**

- Click on any menu item
- Scroll to reviews section
- Enter your name
- Select rating (1-5 stars)
- Write your review text
- Click "Submit Review"

2. **View Reviews:**

- Reviews appear below each menu item

- See rating, reviewer name, and review text
- Reviews are sorted by newest first

Using Sales Dashboard (Admin Only)

1. **Access Sales Dashboard:**

   - Login as admin
   - Sales data is automatically tracked
   - View total revenue, items sold, etc.

2. **View Sales by Item:**

   - See which menu items sell the most
   - See revenue per item

3. **View Sales by Date:**

   - See daily sales totals
   - Track performance over time

Using Profile Features

1. **View Profile:**

   - Click "Profile" button in the header
   - See your account information
   - See your name, email, and role

2. **Logout:**

   - Click "Logout" button
   - You'll be logged out and redirected to login page

Deployment Instructions

Local Deployment (Running on Your Computer)

You've already done this in the setup steps! The application runs on `http://localhost:3000` when you run `node server.js`.

Deploying to Vercel (Online)

Vercel is a platform that lets you host your website online for free.

Step 1: Create a Vercel Account

1. Go to https://vercel.com
2. Sign up for a free account (you can use GitHub, Google, or email)

Step 2: Install Vercel CLI

1. Open your terminal
2. Install Vercel CLI globally:

```
npm install -g vercel
```

Step 3: Deploy

1. Make sure you're in the project folder
2. Run:

```
vercel
```

3. Follow the prompts:
     - Login to Vercel (if not already logged in)
     - Confirm project settings
     - Deploy!

Step 4: Access Your Live Site

1. After deployment, Vercel will give you a URL
2. Your site will be live at that URL
3. Example: https://restora-abc123.vercel.app

Alternative: Deploy via Vercel Website

1. Go to https://vercel.com/dashboard
2. Click "Add New Project"
3. Import your GitHub repository (if you have one)
4. Or drag and drop your project folder
5. Click "Deploy"

Important Notes for Deployment

- The project already has a `vercel.json` file configured
- The database uses file storage (works on Vercel)
- Make sure all your files are uploaded
- Test the live site after deployment

Troubleshooting

Problem: "node: command not found"

Solution:

- Node.js is not installed or not in your PATH

- Install Node.js from https://nodejs.org/
- Restart your terminal after installation

Problem: "Cannot find module" error

Solution:

- Make sure you're in the correct project folder
- Run `npm install` to install dependencies
- Check that all files are present

Problem: "Port 3000 already in use"

Solution:

- Another application is using port 3000
- Option 1: Stop the other application
- Option 2: Change the port in `server.js`:

```
const PORT = process.env.PORT || 3001; // Change 3000 to 3001
```

- Then access at `http://localhost:3001`

Problem: "Cannot access localhost:3000"

Solution:

- Make sure the server is running (check terminal for "Server running" message)
- Check that you typed the URL correctly: `http://localhost:3000`
- Try `http://127.0.0.1:3000` instead
- Check your firewall settings

Problem: "Admin login not working"

Solution:

- Make sure you're using the correct credentials:
  - Email: `admin` or `admin@gmail.com`
  - Password: `admin123456`
- Check browser console for errors (Press F12)
- Clear browser cache and try again

Problem: "Menu items not showing"

Solution:

- Check if the server is running
- Check browser console for errors (Press F12)
- Make sure the API is working: Visit `http://localhost:3000/api/menu`

- You should see JSON data

Problem: "Database errors"

Solution:

- The database uses file storage
- Make sure the `data` folder exists (it will be created automatically)
- Check file permissions
- On Vercel, the database uses `/tmp` folder

Problem: "Images not loading"

Solution:

- Check image URLs in the menu items
- Make sure image URLs are accessible
- Use absolute URLs (starting with `http://` or `https://`)
- Check browser console for 404 errors

Getting More Help

1. **Check the browser console:**

   - Press F12 in your browser
   - Look at the "Console" tab for error messages

2. **Check the server terminal:**

   - Look at the terminal where you ran `node server.js`
   - Error messages will appear there

3. **Check the code:**

   - Make sure all files are present
   - Check for typos in file names

Conclusion

Congratulations! You have successfully set up and learned how to use the Restora Restaurant Management System.

What You've Learned

- How to set up a Node.js web application
- How to run a local server
- How to use all the features of Restora
- How to deploy to Vercel

Next Steps

- Customize the menu with your own items
- Add your own images
- Modify the styling to match your brand
- Add more features as needed

Project Summary

Restora is a complete restaurant management system that includes:

- User authentication
- Menu management
- Order processing
- Table reservations
- Review system
- Sales tracking
- Admin dashboard

The system is built with:

- **Frontend**: HTML, CSS, JavaScript
- **Backend**: Node.js
- **Database**: File-based JSON storage
- **Deployment**: Vercel-ready

Final Notes

- This is a demo project suitable for learning and portfolio
- For production use, consider adding:
  - Real database (PostgreSQL, MongoDB)
  - Real email service (SendGrid, AWS SES)
  - Payment gateway integration
  - User authentication improvements
  - Security enhancements

Thank you for using Restora! Enjoy managing your restaurant! 🍽