Food Ordering Web Application

Technical Documentation Summary

Version 1.0.0
2025-12-09

Project Team

Adham Moatez Mohamed Abd El Monem	22010467
Mohamed Ahmed Agamy	2206164
Marwan Mahmoud Ragab	20221460241
Hazem Ahmed Abd Elfattah Saad	2203175
Youssef Mohamed Helmy Ahmed	2203163

جامعة الإسكندرية
ALEXANDRIA
UNIVERSITY
كلية الحاسبات وعلوم البيانات

Table of Contents

1 Executive Summary 3
1.1 Key Features 3
2 Problem Statement 3
2.1 Challenges Addressed 3
2.2 Solution 3
3 Project Objectives 3
3.1 Primary Objectives 3
3.2 Secondary Objectives 3
4 Technology Stack 4
5 System Architecture 5
5.1 Three-Tier Architecture 5
6 Entity Relationship Diagram 6
7 Mapping Database 6
8 API Endpoints Summary 7
8.1 Authentication (`/api/auth`) 7
8.2 Menu (`/api/menu`) 7
8.3 Orders (`/api/orders`) 7
8.4 Other Endpoints 7
9 User Roles & Permissions 8
10 Order Lifecycle 9
10.1 Status Flow Diagram 9
10.2 Process Steps 9
11 Authentication Flow 10
11.1 Registration Process 10
11.2 Login Process 10
12 Security Features 11
13 Testing 12
13.1 Test Coverage 12
13.2 Running Tests 12
14 Installation & Deployment 12
14.1 Quick Setup 12
14.2 Deployment Options 12
14.3 Feature Completion 12
15 Future Enhancements 13
15.1 Planned Features 13
15.2 Technical Improvements 13
16 Appendices 14
16.1 NPM Scripts 14
17 Conclusion 15

1 Executive Summary

A full-stack MERN (MongoDB, Express, React, Node.js) food ordering platform with role-based access control for Customers, Administrators, and Delivery Personnel.

1.1 Key Features

- **Role-Based Access**: Three user roles with specific permissions
- **Real-Time Tracking**: Live order status updates
- **Admin Dashboard**: Complete system management and analytics
- **PDF Reports**: Sales analytics and reporting
- **Review System**: Customer feedback (1-5 stars)
- **Secure Auth**: JWT-based with bcrypt hashing

2 Problem Statement

2.1 Challenges Addressed

Customer Pain Points:

- Limited accessibility (physical visit/phone required)
- No real-time order visibility
- Restricted payment options
- No structured feedback mechanism

Restaurant Challenges:

- Manual, error-prone order management
- Difficulty tracking inventory
- Inefficient delivery coordination
- Lack of business analytics

Delivery Personnel Challenges:

- Poor communication channels
- Limited order visibility
- Manual status updates

2.2 Solution

A comprehensive web platform bridging customers, restaurants, and delivery personnel with real-time tracking, analytics, and streamlined operations.

3 Project Objectives

3.1 Primary Objectives

	Objective	1
Streamline Ordering	2	Real-Time Tracking
3	Role-Based Access	4
Admin Dashboard	5	Optimize Delivery

3.2 Secondary Objectives

Objective
Customer Engagement

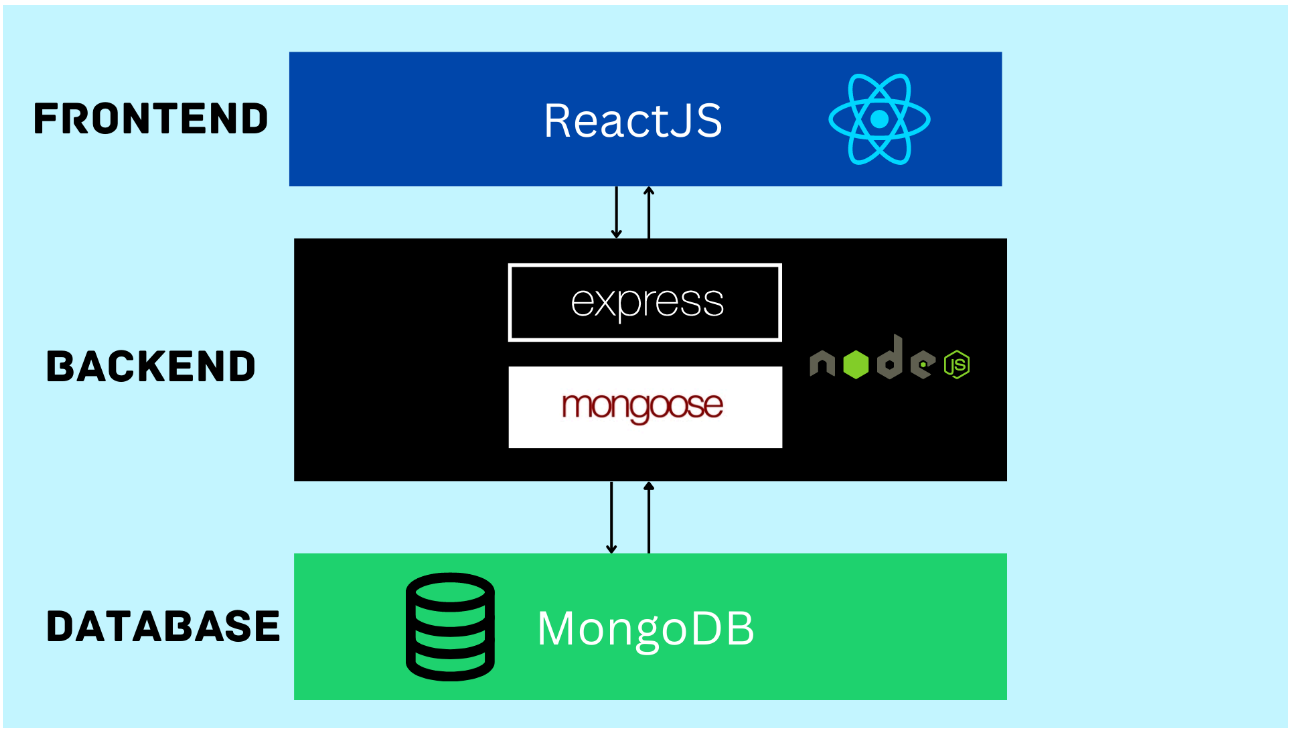
Role Based Access Control
soomthy UI
Tracking order
System Security
Scalability
Data Analytics

4 Technology Stack

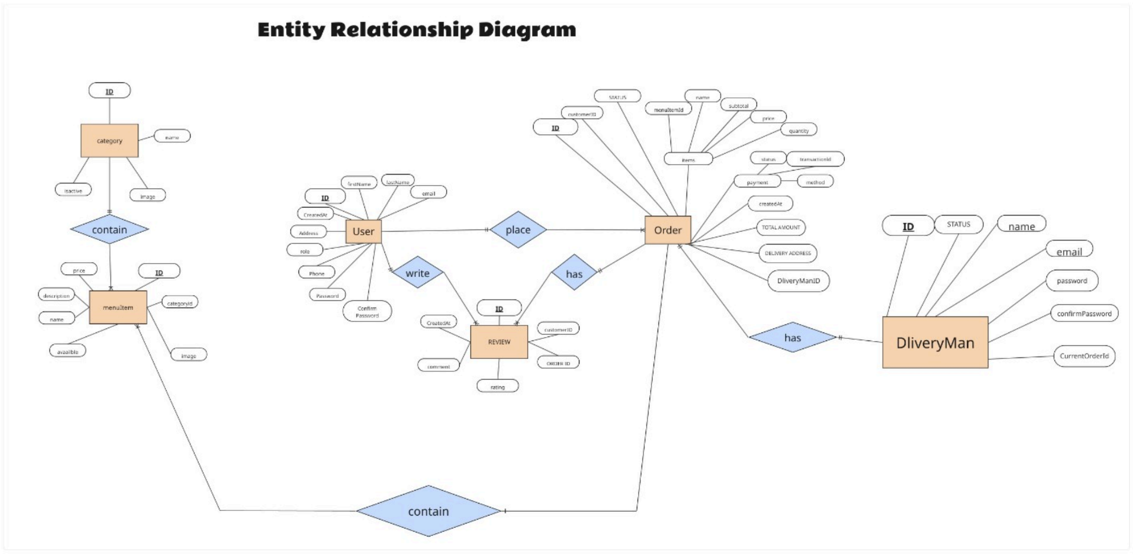
Layer	**Technology**	**Purpose**
Frontend	React 19.2.0	UI library
	Vite 7.2.4	Build tool
	React Router 7.9.6	Routing
	Axios 1.13.2	HTTP client
Backend	Node.js + Express 5.1.0	Server framework
	MongoDB + Mongoose 9.0.0	Database
	JWT 9.0.2	Authentication
	Bcryptjs 3.0.3	Password hashing
	Multer 2.0.2	File uploads
	PDFKit 0.17.2	PDF generation
Testing	Jest 30.2.0	Test framework
	Supertest 7.1.4	HTTP testing

5 System Architecture

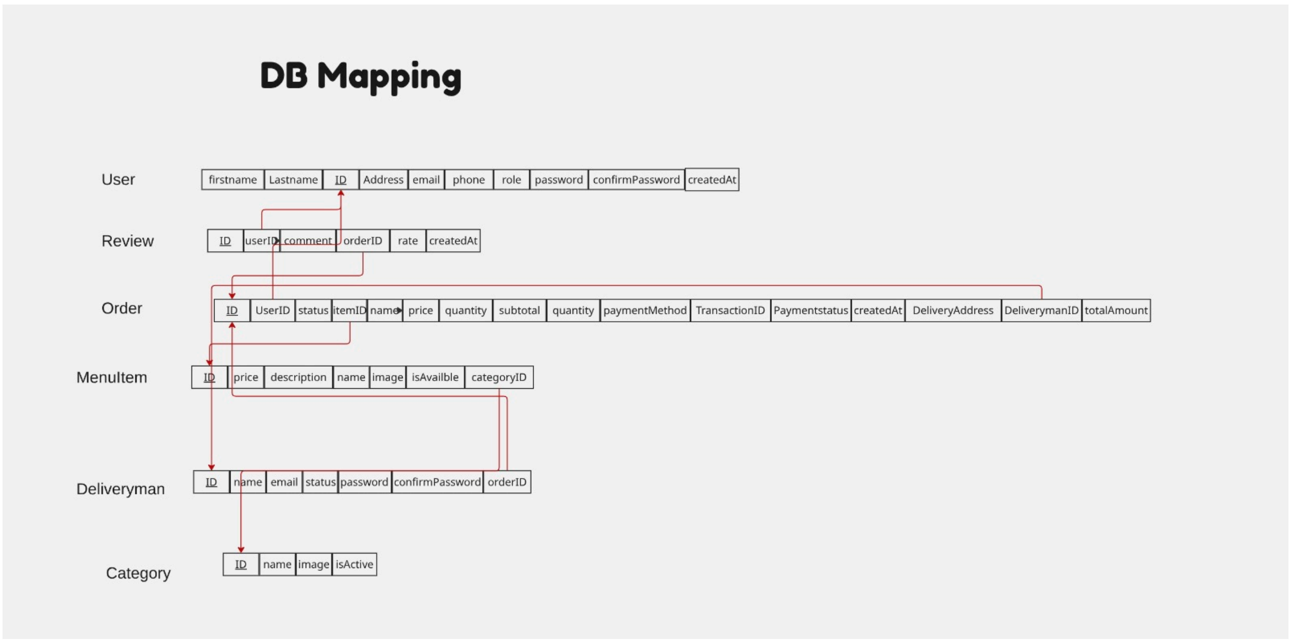
5.1 Three-Tier Architecture



6 Entity Relationship Diagram



7 Mapping Database



User	firstname	Lastname	ID	Address	email	phone	role	password	confirmPassword	createdAt						
Review	ID	userID	comment	orderID	rate	createdAt										
Order	ID	UserID	status	itemID	name	price	quantity	subtotal	quantity	paymentMethod	TransactionID	Paymentstatus	createdAt	DeliveryAddress	DeliverymanID	totalAmount
MenuItem	ID	price	description	name	image	isAvailble	categoryID									
Deliveryman	ID	name	email	status	password	confirmPassword	orderID									
Category	ID	name	image	isActive												

8 API Endpoints Summary

8.1 Authentication (`/api/auth`)

Method	Endpoint	Description
POST	`/register`	Register new user
POST	`/login`	Authenticate user
GET	`/profile`	Get user profile (Protected)
PUT	`/make-admin`	Promote to admin (Dev only)

8.2 Menu (`/api/menu`)

Method	Endpoint	Description
GET	`/`	Get all menu items
POST	`/`	Create menu item (Admin)
PUT	`/:id`	Update menu item (Admin)
DELETE	`/:id`	Delete menu item (Admin)

8.3 Orders (`/api/orders`)

Method	Endpoint	Description
GET	`/`	Get user's orders
POST	`/`	Create new order
PUT	`/:id/status`	Update order status

8.4 Other Endpoints

Route	Endpoints
`/api/categories`	GET, POST (2 endpoints)
`/api/reviews`	GET /:menuItemId, POST (2 endpoints)
`/api/delivery`	POST /register, GET /orders, PUT /accept/:orderId (3 endpoints)
`/api/reports`	GET /dashboard, GET /sales-pdf (2 endpoints)

Total: 25+ API endpoints

9 User Roles & Permissions

Feature	**User**	**Admin**	**Delivery**
Browse Menu	✓	✓	×
Place Orders	✓	✓	×
Track Orders	✓	✓	×
Write Reviews	✓	✓	×
Manage Menu Items	×	✓	×
Manage Categories	×	✓	×
View All Orders	×	✓	×
Generate Reports	×	✓	×
Manage Delivery Personnel	×	✓	×
Accept Deliveries	×	×	✓
Update Delivery Status	×	×	✓

10 Order Lifecycle

10.1 Status Flow Diagram

```
┌─────────┐
│ pending │ ← Customer places order
└────┬────┘
     │
     ▼
┌──────────┐
│Preparing │ ← Admin/System starts preparation
└────┬─────┘
     │
     ▼
┌──────────┐
│onTheWay  │ ← Delivery person accepts & starts
└────┬─────┘
     │
     ▼
┌──────────┐
│delivered │ ← Delivery person completes
└──────────┘
```

10.2 Process Steps

Step	**Action**	**Status**
1	Customer reviews cart and checks out	pending
2	Admin/System starts food preparation	Preparing
3	Delivery person accepts order	onTheWay
4	Delivery person marks as complete	delivered

11 Authentication Flow

11.1 Registration Process

```
1- User fills form → Frontend validates → POST /api/auth/register

2- Backend validates → Hash password (bcrypt) → Save to DB

3- Generate JWT token → Return token + user data

4- Store in localStorage → Redirect to menu
```

11.2 Login Process

```
1- User enters credentials → POST /api/auth/login

2- Find user by email → Compare password (bcrypt)

3- Generate JWT token → Return token + user data

4- Store in localStorage → Update AuthContext

5- Redirect based on role:
  - User → /menu
  - Admin → /admin
  - DeliveryMan → /delivery
```

12 Security Features

Feature	**Implementation**
Password Security	Bcrypt hashing with salt rounds
Authentication	JWT tokens with secret key
Authorization	Role-based middleware
Input Validation	Mongoose schema validation
File Upload	Type and size restrictions (Multer)
CORS	Configured for cross-origin requests
API Protection	Token verification middleware

13 Testing

13.1 Test Coverage

Test Suite	Tests	Coverage
auth.controller.test.js	7	Register, Login, GetProfile
delivery.controller.test.js	6	Register, Orders, Accept
menu.controller.test.js	8	CRUD operations
order.controller.test.js	7	Create, Get, Update
review.controller.test.js	5	Create, Get Reviews
report.controller.test.js	6	Dashboard, PDF

Total: 39 passing tests

13.2 Running Tests

```
cd backend
npm test
```

14 Installation & Deployment

14.1 Quick Setup

```
# 1. Install dependencies
npm run install-all

# 2. Configure .env (backend/)
PORT=5000
Mongo_URL=mongodb://localhost:27017/food_ordering_db
JWT_SECRET=your_secret_key

# 3. Run application
npm run dev

# Access:
# Frontend: http://localhost:5173
# Backend: http://localhost:5000
```

14.2 Deployment Options

Option	Platform	Command
1	Vercel (Recommended)	`vercel --prod`
2	Docker	Containerize and deploy
3	Traditional	Heroku, Railway, DigitalOcean

14.3 Feature Completion

- User authentication & authorization
- Menu browsing & management
- Shopping cart functionality
- Order placement & tracking
- Review & rating system
- Admin dashboard

- Delivery management
- PDF report generation
- File upload system
- Comprehensive testing

15 Future Enhancements

15.1 Planned Features

Phase	**Features**
Phase 1	Real-time notifications (WebSockets), Push notifications, Advanced search
Phase 2	Payment integration (Stripe, PayPal), Wallet system, Promo codes
Phase 3	Multi-restaurant support, Table reservations, Loyalty program
Phase 4	Mobile app (React Native), GPS tracking, Offline mode

15.2 Technical Improvements

- Redis caching
- Rate limiting
- API versioning
- Comprehensive logging
- Automated backups
- Monitoring & alerting
- CI/CD pipeline
- End-to-end testing
- GraphQL API
- Internationalization (i18n)

16 Appendices

16.1 NPM Scripts

Root Package:

- `npm run install-all` - Install all dependencies
- `npm run dev` - Run frontend + backend
- `npm run vercel-build` - Build for Vercel

Backend:

- `npm start` - Production server
- `npm run dev` - Development with watch
- `npm test` - Run tests

Frontend:

- `npm run dev` - Vite dev server
- `npm run build` - Production build
- `npm run preview` - Preview build

17 Conclusion

This Food Ordering Web Application successfully addresses traditional food ordering challenges through:

- **Modern Technology Stack**: MERN stack with best practices
- **Comprehensive Features**: Complete ordering, tracking, and management system
- **Security**: JWT authentication, bcrypt hashing, role-based access
- **Testing**: 39 passing tests ensuring reliability
- **Documentation**: Complete technical and API documentation

The platform provides an efficient, user-friendly solution for customers, restaurants, and delivery personnel, with room for future enhancements and scalability.

—

Developed By:

- Adham Moatez Mohamed Abd El Monem (22010467)
- Mohamed Ahmed Agamy (2206164)
- Marwan Mahmoud Ragab (20221460241)
- Hazem Ahmed Abd Elfattah Saad (2203175)
- Youssef Mohamed Helmy Ahmed (2203163)