

Queue Hero - Complete Project Structure

```
queue-hero/
├── package.json
├── server.js      # Main Express server entry point
├── config/
│   ├── database.js    # Database configuration and connection
│   ├── auth.js        # JWT and authentication settings
│   └── notifications.js  # Twilio and notification settings
├── middleware/
│   ├── auth.js        # JWT verification middleware
│   ├── validation.js   # Request validation middleware
│   └── accessibility.js # WCAG compliance helpers
├── models/
│   ├── User.js        # Customer and Agent user models
│   ├── Queue.js       # Queue request data model
│   ├── Store.js       # Store/location information model
│   └── Notification.js # Notification tracking model
├── routes/
│   ├── auth.js        # Authentication endpoints
│   ├── queues.js      # Queue CRUD operations
│   ├── users.js       # User management
│   ├── stores.js      # Store management
│   └── webhooks.js    # WhatsApp webhook handlers
├── services/
│   ├── queueService.js # Core queue business logic
│   ├── notificationService.js # SMS/WhatsApp/Push notifications
│   ├── etaService.js    # Dynamic ETA calculations
│   └── whatsappService.js # WhatsApp integration logic
├── sockets/
│   ├── queueUpdates.js # Socket.IO queue event handlers
│   └── agentDashboard.js # Real-time agent communication
├── database/
│   ├── migrations/     # Database schema migrations
│   │   ├── 001_initial_schema.sql
│   │   ├── 002_add_notifications.sql
│   │   └── 003_add_accessibility_features.sql
│   ├── seeds/          # Sample data for development
│   │   ├── stores.sql
│   │   └── test_users.sql
│   └── schema.sql      # Complete database schema
├── public/            # Static frontend assets
│   ├── index.html      # Customer interface
│   ├── agent-dashboard.html # Agent/Hero interface
│   ├── admin-panel.html # Store manager interface
│   └── css/
│       ├── main.css     # Core styles with grey-gold-black theme
│       └── accessibility.css # WCAG compliance styles
```

```
|- themes.css      # Dark/light theme switching
|- components.css # Reusable UI components
|- js/
|   |- app.js       # Main customer application logic
|   |- agent-dashboard.js # Agent interface functionality
|   |- admin-panel.js # Admin interface logic
|   |- socket-client.js # Socket.IO client connection
|   |- accessibility.js # Keyboard navigation and ARIA helpers
|   |- theme-switcher.js # Dark/light mode toggle
|   |- utils.js     # Shared utility functions
|- images/
|   |- logo.png
|   |- hero-icon.svg
|   |- accessibility-icons/
|- manifest.json    # PWA manifest for offline capabilities
|- tests/
|   |- unit/
|   |   |- services/
|   |   |- models/
|   |   |- routes/
|   |- integration/
|   |   |- queue-flow.test.js
|   |   |- notification.test.js
|   |- accessibility/
|   |   |- keyboard-nav.test.js
|   |   |- screen-reader.test.js
|- docs/
|   |- API.md        # API documentation
|   |- ACCESSIBILITY.md # WCAG compliance documentation
|   |- DEPLOYMENT.md    # Deployment instructions
|   |- WHATSAPP_SETUP.md # WhatsApp integration guide
|- .env.example      # Environment variables template
|- .gitignore
|- README.md
```

Key Architecture Decisions

Database Design Philosophy

- **SQLite for MVP**: Single-file database perfect for initial deployment and testing
- **Migration-ready**: Structured to easily transition to PostgreSQL/MySQL for scaling
- **ACID compliance**: Ensures queue position integrity during concurrent operations

Frontend Architecture

- **Vanilla JS**: No framework dependencies, faster load times, better accessibility control
- **Progressive Enhancement**: Works without JavaScript, enhanced with it
- **Component-based CSS**: Reusable styles following BEM methodology
- **Socket.IO Integration**: Real-time updates without page refreshes

Backend Service Layer

- **Service-oriented**: Business logic separated from route handlers
- **Event-driven**: Queue updates trigger cascading notifications
- **Middleware-heavy**: Authentication, validation, and accessibility checks at every layer

Accessibility-First Design

- **WCAG 2.1 AA compliance**: Built into every component from the ground up
- **Keyboard navigation**: Tab order and focus management prioritized
- **Screen reader optimization**: Semantic HTML with comprehensive ARIA labels
- **High contrast support**: Color schemes tested for visibility impairments

Real-time Communication Strategy

- **Socket.IO rooms**: Separate channels for customers, agents, and admins
- **Event segregation**: Different update types for queue position, ETA changes, agent status
- **Fallback polling**: Graceful degradation when WebSocket connections fail