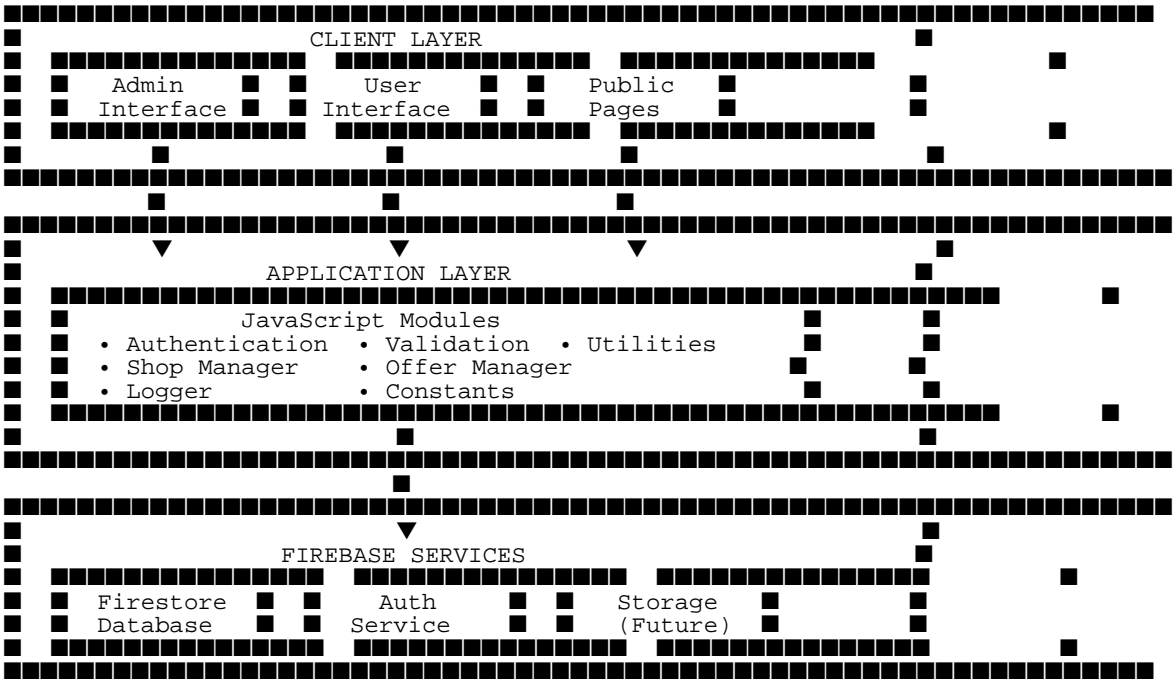


# Super Mall Web Application

## Project Overview

A comprehensive web application for managing and browsing a Super Mall, allowing administrators to manage shops, offers, categories, and floors, while providing users with an intuitive interface to explore shops, compare products, and discover exclusive offers.

## System Architecture



## Database Schema (Firestore)

### Collections:

**\*\*shops\*\***

- name (string)
- description (string)

- category (string)
- floor (string)
- contact (string)
- email (string)
- status (string: active/inactive/pending)
- createdAt (timestamp)
- updatedAt (timestamp)

#### **\*\*offers\*\***

- title (string)
- description (string)
- shopId (string)
- shopName (string)
- offerType (string: percentage/fixed\_amount/bogo/seasonal)
- discount (number)
- startDate (string)
- endDate (string)
- createdAt (timestamp)
- updatedAt (timestamp)

#### **\*\*categories\*\***

- name (string)
- description (string)
- icon (string)
- createdAt (timestamp)

#### **\*\*floors\*\***

- name (string)
- code (string)
- level (number)

- description (string)
- createdAt (timestamp)

#### **\*\*products\*\*** (Future Enhancement)

- name (string)
- description (string)
- price (number)
- category (string)
- shopId (string)
- images (array)
- features (array)
- createdAt (timestamp)

## Project Structure

```

Super Mall Web Application/
├── index.html                # Landing page
├── admin/                    # Admin module
│   ├── login.html           # Admin login page
│   ├── dashboard.html       # Admin dashboard
│   ├── shops.html           # Shop management
│   ├── offers.html          # Offer management
│   ├── categories.html      # Category management
│   └── floors.html          # Floor management
├── user/                     # User module
│   ├── shops.html           # Browse shops
│   ├── offers.html          # Browse offers
│   ├── categories.html      # Category-wise browsing
│   ├── floors.html          # Floor-wise browsing
│   └── shop-details.html    # Individual shop details
├── assets/                   # Static assets
│   ├── css/
│   │   ├── common.css      # Common styles
│   │   ├── home.css        # Home page styles
│   │   ├── admin.css       # Admin panel styles
│   │   └── user.css         # User interface styles
│   └── js/
│       ├── admin/
│       │   ├── login.js     # Admin authentication
│       │   ├── auth-check.js # Auth middleware
│       │   ├── dashboard.js # Dashboard logic
│       │   ├── shops.js     # Shop management
│       │   ├── offers.js    # Offer management
│       │   ├── categories.js # Category management
│       │   └── floors.js    # Floor management
│       └── user/
│           ├── shops.js     # Shop browsing
│           └── offers.js    # Offer browsing

```

```

■   ■   ■   ■■■ categories.js # Category browsing
■   ■   ■   ■■■ floors.js    # Floor browsing
■   ■   ■
■   ■   ■■■ home.js          # Home page logic
■   ■
■   ■■■ images/              # Image assets
■   ■■■ favicon.ico
■
■■■ config/                  # Configuration files
■   ■■■ firebase-config.js   # Firebase configuration
■
■■■ utils/                   # Utility modules
■   ■■■ logger.js            # Logging utility
■   ■■■ constants.js         # Application constants
■   ■■■ helpers.js           # Helper functions
■   ■■■ validation.js        # Validation module
■
■■■ docs/                    # Documentation
■   ■■■ LLD.md               # Low-Level Design
■   ■■■ ARCHITECTURE.md       # System Architecture
■   ■■■ WIREFRAMES.md        # Wireframe Documentation
■   ■■■ TEST_CASES.md        # Test Cases
■   ■■■ API_DOCUMENTATION.md # API Documentation
■
■■■ tests/                   # Test files
■   ■■■ unit/                 # Unit tests
■   ■■■ integration/         # Integration tests
■   ■■■ e2e/                 # End-to-end tests
■
■■■ .gitignore               # Git ignore file
■■■ README.md                # This file
■■■ LICENSE                  # License file

```

## Testing

### Manual Testing

See `docs/TEST\_CASES.md` for comprehensive test cases covering:

- Admin authentication
- Shop CRUD operations
- Offer management
- Category management
- Floor management
- User browsing features
- Filter and search functionality
- Data validation
- Error handling

# Code Standards

## JavaScript

- ES6+ syntax
- Modular architecture
- JSDoc comments for functions
- Consistent naming conventions (camelCase)
- Error handling for all async operations
- Logging for all major actions

## HTML

- Semantic HTML5 elements
- Accessibility attributes (ARIA labels)
- Clean indentation (2 spaces)
- Descriptive IDs and classes

## CSS

- BEM naming convention
- CSS variables for theming
- Mobile-first responsive design
- Organized by component

## Security

- Input sanitization
- XSS protection
- CSRF protection
- Secure authentication
- Environment variable protection

# Optimization

## Code Level

- Modular functions
- Reusable components
- Efficient algorithms
- Minimal DOM manipulation

## Database Level

- Indexed queries
- Batch operations
- Optimized security rules
- Minimal read/write operations

# Acknowledgments

- Firebase for backend services
- MDN Web Docs for web standards
- Font Awesome for icons (if used)
- Community contributors

# Future Enhancements

- ☐ Product management module
- ☐ Shopping cart functionality
- ☐ User authentication and profiles
- ☐ Payment integration

- ☐ Push notifications
- ☐ Analytics dashboard
- ☐ Mobile app (React Native/Flutter)
- ☐ Advanced search with Elasticsearch
- ☐ Recommendation engine
- ☐ Multi-language support

**\*\*Note\*\*:** Replace placeholder values (YOUR\_API\_KEY, your.email@example.com, etc.) with actual values before deployment.