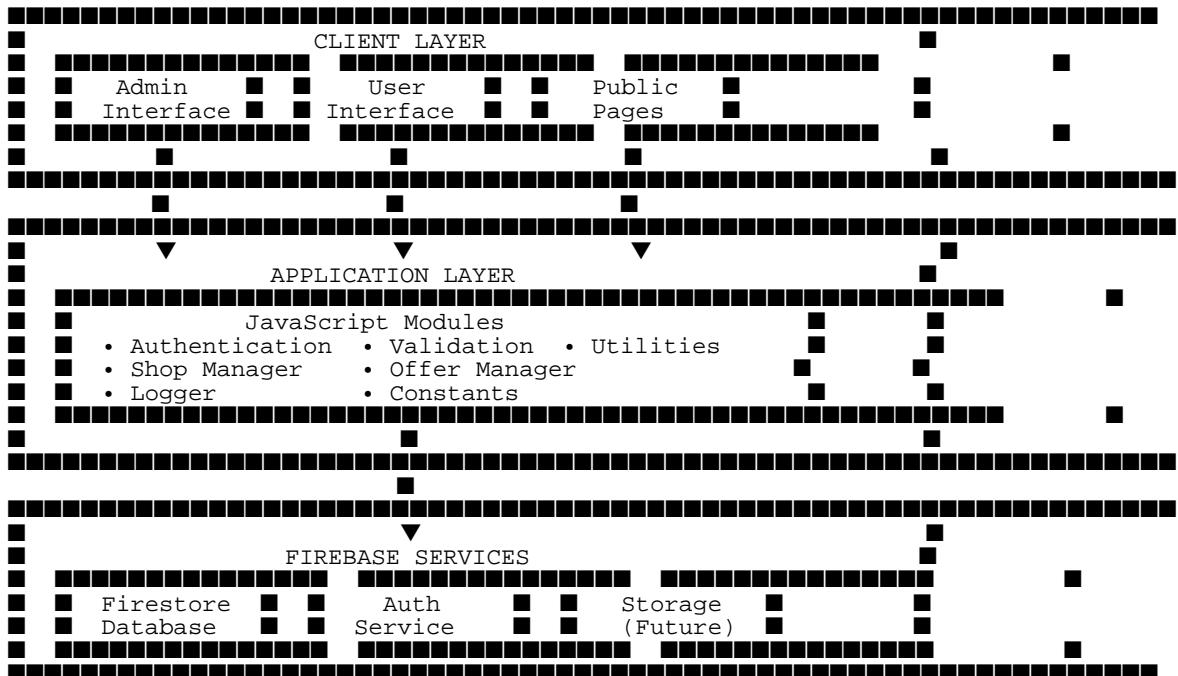


# Super Mall Web Application

## 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/
        login.html                  # Admin module
        dashboard.html              # Admin login page
        shops.html                  # Admin dashboard
        offers.html                 # Shop management
        categories.html             # Offer management
        floors.html                 # Category management
        floors.html                 # Floor management
    user/
        shops.html                  # User module
        offers.html                 # Browse shops
        categories.html             # Browse offers
        floors.html                 # Category-wise browsing
        floors.html                 # Floor-wise browsing
        shop-details.html           # Individual shop details
    assets/
        css/
            common.css               # Static assets
            home.css                 # Common styles
            admin.css                # Home page styles
            user.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               # User interface
                offers.js              # Offer browsing

```

```
    categories.js # Category browsing
    floors.js     # Floor browsing
    home.js       # Home page logic
    images/
        favicon.ico # Image assets

    config/
        firebase-config.js # Configuration files
                            # Firebase configuration

    utils/
        logger.js      # Utility modules
        constants.js   # Logging utility
        helpers.js     # Application constants
        validation.js  # Helper functions
                            # Validation module

    docs/
        LLD.md         # Documentation
        ARCHITECTURE.md # Low-Level Design
        WIREFRAMES.md  # System Architecture
        TEST_CASES.md   # Wireframe Documentation
        API_DOCUMENTATION.md # Test Cases
                            # API Documentation

    tests/
        unit/          # Test files
        integration/   # Unit tests
        e2e/           # Integration tests
                            # End-to-end tests

    .gitignore
    README.md
    LICENSE
```

## 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.