

ਮੀਠੋ ਗਾਸ - Complete Documentation

Welcome to the complete documentation for Mithogaas! This guide will help you understand every aspect of the application.

Table of Contents

ਮੀਠੋ ਗਾਸ - Complete Documentation	1
Getting Started	2
Installation Steps	2
Environment Setup	2
Architecture Overview	2
Application Flow	2
Core Technologies	2
Features Deep Dive	3
1. Recipe Search & Discovery	3
2. Favorites System	3
3. Meal Planning	3
4. Recipe Comparison	4
5. Recipe Notes & Customization	4
6. Statistics Dashboard	5
Component Reference	6
Pages	6
Components	7
State Management	8
Context Providers	8
API Integration	8
TheMealDB API	8
Storage System	8
localStorage Keys	8
Security Considerations	8
localStorage Security	9
Support	9
Credits	9
APIs & Services	9
Tools & Libraries	9
Inspiration	9

Getting Started

Installation Steps

```
# Clone the repository  
git clone https://github.com/yourusername/mithogaas.git
```

```
# Navigate to project directory  
cd mithogaas
```

```
# Install dependencies  
npm install
```

```
# Start development server  
npm run dev
```

Environment Setup

No environment variables are required! The app uses:

- TheMealDB free API (no API key needed)
- localStorage for data persistence
- No backend server required

Architecture Overview

Application Flow

```
User → React Router → Pages → Components → Hooks → localStorage  
                                ↓  
                                TheMealDB API
```

Core Technologies

Frontend Framework:

- React 18 with hooks
- Functional components
- Context API for state

Routing:

- React Router v6
- Client-side navigation
- Protected routes

Styling:

- Tailwind CSS utility classes
- Custom CSS animations
- Responsive design

Data Management:

- localStorage for persistence
 - React Context for global state
 - Custom hooks for logic
-

Features Deep Dive

1. Recipe Search & Discovery

How it works:

1. User types in search bar
2. Debounced search (500ms delay)
3. API call to TheMealDB
4. Results displayed with loading skeleton
5. Filter by categories
6. Sort alphabetically

Key Components:

- SearchBar.jsx - Input with debouncing
- RecipeCard.jsx - Individual recipe display
- Skeletons.jsx - Loading states

2. Favorites System

Storage Structure:

```
localStorage.favorites = [
```

```
{  
  idMeal: "52772",  
  strMeal: "Teriyaki Chicken",  
  strCategory: "Chicken",  
  strArea: "Japanese",  
  strMealThumb: "....."  
,  
..... more recipes
```

```
]
```

Key Features:

- Add/remove recipes
- Persistent across sessions
- Quick access page
- Badge indicators

3. Meal Planning

Storage Structure:

```
localStorage.mealPlan = {
```

```
  Monday: [recipe1, recipe2],  
  Tuesday: [recipe3],  
  Wednesday: [],  
  ... rest of week
```

```
}
```

Features:

- Weekly calendar view
- Assign recipes to days
- Remove individual recipes
- Clear entire day or week
- Generate shopping list

```

Shopping List Generation:
const generateShoppingList = () => {
  const allRecipes = getAllRecipes();
  const ingredientMap = { };

  allRecipes.forEach(recipe => {
    // Extract and aggregate ingredients
    for (let i = 1; i <= 20; i++) {
      const ingredient = recipe[`strIngredient${i}`];
      const measure = recipe[`strMeasure${i}`];
      // Combine duplicates
    }
  });
}

return sortedIngredients;
};

```

4. Recipe Comparison

How it works:

1. User clicks "Add to Compare" on recipe cards
2. Floating bar appears at bottom
3. Select 2-3 recipes
4. Navigate to comparison page
5. View side-by-side details

Comparison Data:

- Category, Cuisine, Ingredient count
- Cooking steps count
- Estimated cooking time
- Common ingredients highlighted
- Full ingredient lists
- Step-by-step instructions

Storage:

```

localStorage.comparisonList = [recipe1, recipe2, recipe3]
// Max 3 recipes

```

5. Recipe Notes & Customization

Storage Structure:

```

localStorage.recipeNotes = {
  "52772": {
    notes: [
      {
        id: 1234567890,
        text: "Added extra garlic",
        date: "2026-01-07T12:00:00",
        edited: false
      }
    ],
    modifications: [
      {
        ...
      }
    ]
  }
}

```

```
        id: 1234567891,
        original: "butter",
        substitute: "olive oil",
        reason: "healthier option",
        date: "2026-01-07T12:00:00"
    }
],
rating: 4,
cookingHistory: [
{
    id: 1234567892,
    date: "2026-01-07T12:00:00",
    success: true,
    comments: "Turned out great!"
}
],
tips: [
{
    id: 1234567893,
    text: "Reduce cooking time by 5 mins",
    date: "2026-01-07T12:00:00"
}
]
}
```

Features:

- Personal notes (unlimited)
- Ingredient substitutions
- Star ratings (1-5)
- Cooking session logs
- Success rate tracking
- Personal tips

6. Statistics Dashboard

Tracked Metrics:

Overview:

- Total recipes viewed (unique)
- Total favorites
- Total planned meals
- Total customized recipes

Cooking Stats:

- Times cooked
- Success rate (%)
- Recent views (7 days)

Visual Charts:

- Category breakdown (bar chart)
- Cuisine exploration (bar chart)
- Activity heatmap (30 days)
- Top 5 most viewed

- Top 5 rated recipes

Viewing History:

```
localStorage.viewingHistory = [
  {
    recipeId: "52772",
    recipeName: "Teriyaki Chicken",
    category: "Chicken",
    area: "Japanese",
    image: ".....",
    timestamp: "2026-01-07T12:00:00"
  },
  // Last 100 views stored
]
```

Component Reference

Pages

HomePage.jsx

- Hero section with banner
- Feature highlights
- Statistics overview
- Testimonials

Recipes.jsx

- Search functionality
- Category filters
- Recipe grid
- Sort options
- Empty states

RecipeDetail.jsx

- Full recipe information
- Ingredients list
- Instructions
- Video tutorial
- Action buttons (Favorite, Compare, Meal Plan)
- Notes section

Favorites.jsx

- Saved recipes grid
- Empty state
- Quick actions

MealPlanner.jsx

- Weekly calendar
- Day cards
- Shopping list modal
- PDF export

RecipeComparison.jsx

- Side-by-side comparison

- Tabbed interface
- Common ingredients
- Remove options

StatsDashboard.jsx

- Overview cards
- Charts and graphs
- Activity heatmap
- Top lists

Components

SearchBar.jsx

```
// Debounced search input
const SearchBar = ({ onSearch }) => {
  const [query, setQuery] = useState("");

  useEffect(() => {
    const timer = setTimeout(() => {
      if(query) onSearch(query);
    }, 500);

    return () => clearTimeout(timer);
  }, [query]);

  return <input ... />;
};
```

RecipeCard.jsx

- Recipe thumbnail
- Name, category, cuisine
- "Add to Compare" button
- "Notes" badge
- Link to detail page

ComparisonBar.jsx

- Fixed bottom bar
- Recipe thumbnails
- Remove buttons
- "Compare Now" button

RecipeNotesSection.jsx

- Tabbed interface
- Rating system
- Forms for adding data
- Lists of notes/tips/etc.

Skeletons.jsx

- Loading placeholders
- Multiple variants
- Pulse animations

State Management

Context Providers

1. FavoritesProvider

```
const FavoritesContext = createContext();

export const FavoritesProvider = ({ children }) => {
  const [favorites, setFavorites] = useState(loadFromStorage);

  useEffect(() => {
    saveToStorage(favorites);
  }, [favorites]);

  return (
    <FavoritesContext.Provider value={{...}}>
      {children}
    </FavoritesContext.Provider>
  );
};
```

2. MealPlannerProvider

- Manages weekly meal plan
- Generates shopping lists
- Clears days/weeks

3. ComparisonProvider

- Tracks compared recipes (max 3)
- Add/remove functionality
- Validation logic

4. RecipeNotesProvider

- Manages all recipe customizations
- CRUD operations for notes
- Rating system
- Cooking logs

API Integration

TheMealDB API

Storage System

localStorage Keys

```
{
  "favorites": "[]",
  "mealPlan": "{}",
  "comparisonList": "[]",
  "recipeNotes": "{}",
  "viewingHistory": "[]"
}
```

Security Considerations

localStorage Security

- No sensitive data stored
 - All data is user-generated
 - No passwords or tokens
 - Public API (no authentication)
-

Support

Need help? Here's how to get support:

1. **Check Documentation** - Most answers are here
 2. **LinkedIn** - <https://www.linkedin.com/in/bijay-jha-b0b7a9290/>
-

Credits

APIs & Services

- TheMealDB - Recipe data

Tools & Libraries

- React - UI framework
- Tailwind CSS - Styling
- Vite - Build tool
- React Router – Routing

Inspiration

- Modern recipe apps
 - Food blogging community
 - Home cooking enthusiasts
-