

# Expense Tracker Application

## Full-Stack Web App on React & Supabase

June 2025

### Project Description

---

Expense Tracker is a modern, full-stack web application to efficiently keep track of one's own expenses. Built on React and TypeScript on the client side and Supabase as the backend-as-a-service, the application is a self-contained system to track expenses with secure authentication and real-time data syncing.

### Key Features

---

- **User Auth:** Complete authentication system from register, login, reset password, and update
- **Expense Management:** Add, read, update, and delete expenses with categorization
- **Bulk Entry:** More than one expense entry at once for ease
- **Real-time Summaries:** In-flight grouping categorization and summation
- **Dark Theme UI:** New black UI with blue accent colors
- **Local Storage Fallback:** Offline support when database is offline
- **Type Safety:** Complete TypeScript support for safety-first development
- **Responsive Design:** Mobile-first with cross-device compatibility

### Technical Architecture

---

#### Frontend Technology Stack

- **React 18:** New component architecture focused on hooks
- **TypeScript:** Static types for better code quality and developer experience
- **Vite:** Lightning-fast build tool and dev server
- **CSS3:** Dark theme support with customizable styling

#### Backend & Database

- **Supabase:** Realtime-enabled PostgreSQL database
- **Row Level Security (RLS):** Data protection at the user level
- **PostgREST API:** Automatic REST API generation from database schema
- **Authentication:** JWT-based authentication via email verification

#### Database Schema

Application has a three-table normalized database schema:

- **expenses:** Primary expense details with user id, amount, description, category, and date
- **categories:** Predefined expense category list with color and icon
- **profiles:** User profile details for authentication

## Application Features & Functionality

---

### Authentication System

App includes full authentication system:

- User signup with email verification
- Secure sign-in with JWT token management
- Modal password reset through email sending
- Automatic token renewal and session handling

### Expense Management

**Single Expense Entry:** Single expense input with amount, description, category select, and date select feature.

**Bulk Expense Entry:** Pro feature to allow adding multiple expenses at a time through a modal window with:

- Dynamic add/remove rows
- Field-level validation
- Performance-optimized quick action buttons
- Batch submission with error reporting

**Expense Summary:** Computed and shown in real time

- Total cost across the board
- Breakup category-wise with visual hints
- Day-wise and month-wise spending pattern

## API Integration & Testing

---

### RESTful API Design

The app talks to Supabase through a tidy REST API:

- **Authentication Endpoints:** /auth/v1/ for user management
- **Data Endpoints:** /rest/v1/ for CRUD operations
- **RPC Functions:** Custom DB functions for complex queries

### Postman API Testing

Effortless API test suite includes:

- Full Postman collection with 20 tests
- Environment variables for various deployment stages
- Token management auto authentications

## Security & Data Protection

---

### Security Controls

- **Row Level Security:** Database-level separation of user data
- **JWT Authentication:** Secure session management using tokens
- **Input Validation:** Server-side and client-side validation of data
- **HTTPS Encryption:** All data in transit is encrypted

### Data Privacy

- Single users can only view their expense data only
- Password reset flows with secure token validation
- Fallback to local storage preserves offline privacy
- No sensitive information in browser local storage

## Development & Deployment

---

### Development Workflow

- Type safety and dev productivity via TypeScript
- Component-based design to maintainability
- State management custom hooks (useAuth, useRecoveryMode)
- Service layer abstraction for API invocation

### Code Quality

- Code consistency via ESLint configuration
- Strict TypeScript mode for improved type checking
- Component-level CSS with modular CSS
- Robust error boundaries and proper error handling

## Conclusion

---

The Expense Tracker application is a comprehensive showcase of best practice web development with its backend development in React, TypeScript, and Supabase. The application is a robust, secure, and inviting platform for tracking personal expenses with full authentication, real-time data synchronization, and robust API test coverage support.

The project addresses frontend development, backend integration, database modeling, API development, and integration of security.