

# AI- Campus Assignment - Prathamesh Vasaikar (2022BCS0072)

[prathameshvasaikarat3x4@gmail.com](mailto:prathameshvasaikarat3x4@gmail.com)

Deployed Link : <https://pratham-expense.vercel.app/>  
(username - pratham , password- pratham3x4 )

Github : [https://github.com/Prthm444/Expense\\_Tracker](https://github.com/Prthm444/Expense_Tracker)

## Introduction - Expense-Tracker

Expense-Tracker is a web application designed for creating, and managing your personal expenses. Users can register, log in, post, edit, and delete their expenses. With secure session management, you stay logged in even after refreshing the page for a seamless experience.

### Functional Requirements

- Registration of a user
- Login of a user
- Browse and see expenses added by you
- Editing your expenses
- Deleting your expenses
- Filtering expenses based on categories
- Logging out

### Non-Functional Requirements

- **Session Persistence:** The web app must store access tokens securely (e.g., in local storage or cookies) so that the user's session persists even after a page refresh, ensuring that users do not get logged out upon refreshing.
- **Secure API Access :** The application uses authentication middleware to ensure that only authenticated users can access protected APIs, thereby preventing unauthorized access.

# Schema design for CRUD:

User:

- Each user has a unique ID.
- Users have a username and an email address, both unique.
- Passwords are stored securely as hashes (never plain text).
- Each user record keeps track of when it was created and last updated.

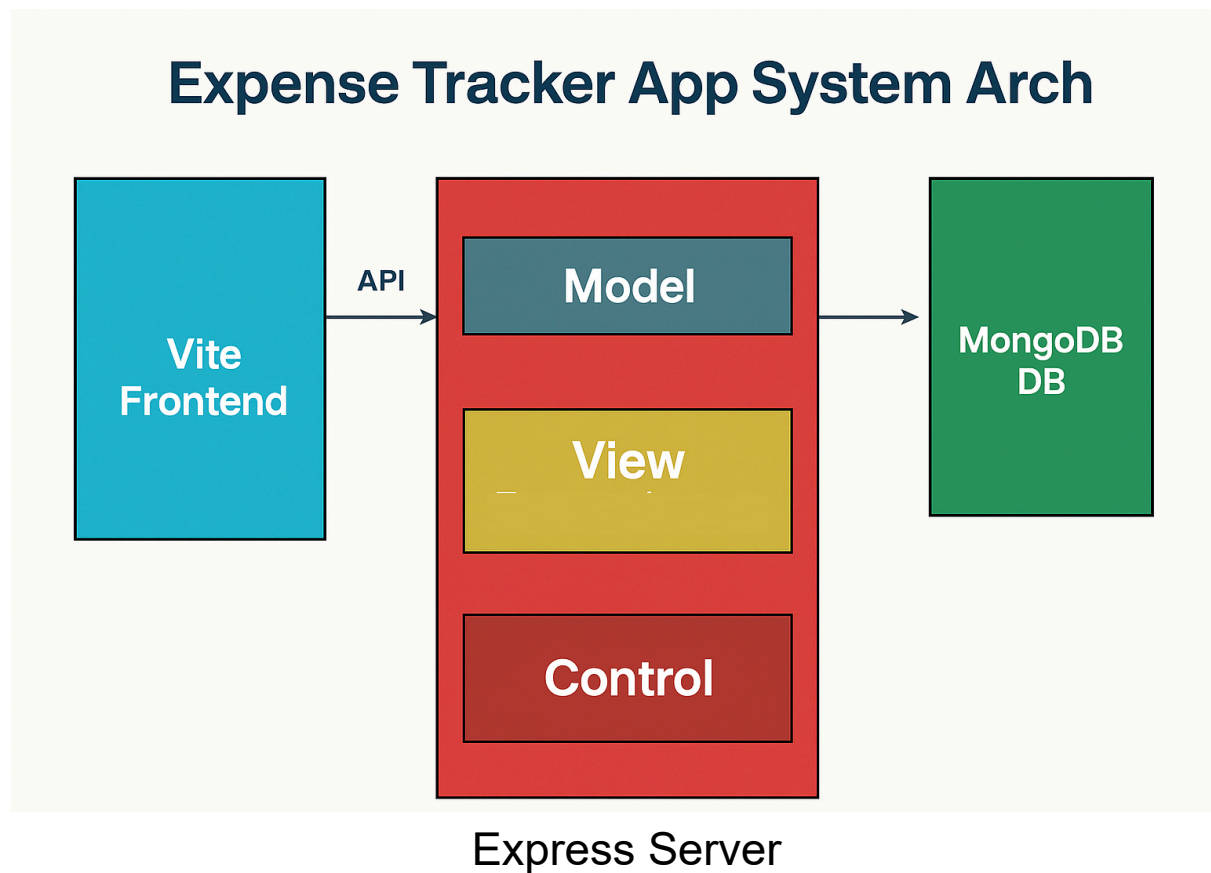
```
{
  _id: ObjectId,          // Automatically generated by MongoDB
  username: String,       // Unique, required
  email: String,          // Unique, required
  password_hash: String,  // Required (hashed password)
  created_at: Date,       // Set when document is created
  updated_at: Date        // Updated when document changes
}
```

Blog:

- Each blog post has a unique ID.
- Blog posts have a title and content.
- Each blog post has an author, which refers to the user who created it.
- Blog posts also track when they were created and last updated.

```
{
  _id: ObjectId,          // Automatically generated by MongoDB
  title: String,          // Required
  category: String,       // Required
  is_recurring: Boolean,   // References _id in users collection
  unitPrice : Integer,     // required
  quantity : Integer,     // must be >0
  totalCost : Integer,    // calculated when saved
  created_at: Date,       // Set when document is created
  updated_at: Date        // Updated when document changes
}
```

## Module and Class structure:



- Express backend (using MVC pattern)
- React frontend
- MongoDB database

# Backend (Express Server — MVC Architecture)

## Modules

- controllers/
  - Handles HTTP requests and application logic.
- models/
  - Defines MongoDB schemas and data operations.
- routes/
  - Maps endpoints to controller methods.
- middlewares/
  - Contains authentication, error handling, etc.
- db/
  - Handles connection to db.
- utils/
  - Utility/helper functions (input validation, hashing).