

Expenses Tracker

Names:

Mohamed Mosaad Shota Abd-elshafy

Abd-elrhman Hussein Mohamed Al-asfar

Khaled Yasser Ali Zidan

Mohamed Mosaad Mohamed Metwaly El-gazar

Under Supervision of
Assist Prof: Hassan El-deeb

Table of contents

01 WEB Design

02 Database

03 Backend

04 Frontend



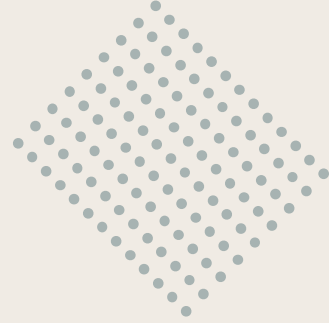
01

Web Design

UI/UX Design and System Design

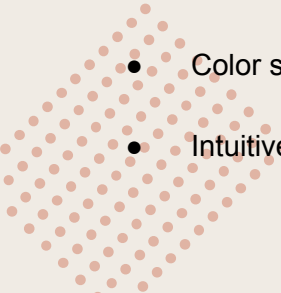


UI/UX Design



- The UI/UX was designed using **Figma** for clear visual structure and consistency.
- The layout focuses on **simplicity** and **ease of navigation** for all user types.
- Key screens include:
 - Dashboard with spending overview
 - Expense & income entry forms
 - Recurring transaction management
 - Budget tracking modules
- The design ensures a **responsive layout** for both desktop and mobile views.
- Color scheme and iconography were chosen to enhance **readability** and **user focus**.
- Intuitive interactions and minimal steps for entering financial data improve the **user experience**.

To Visit our Design [Click Here](#)





02

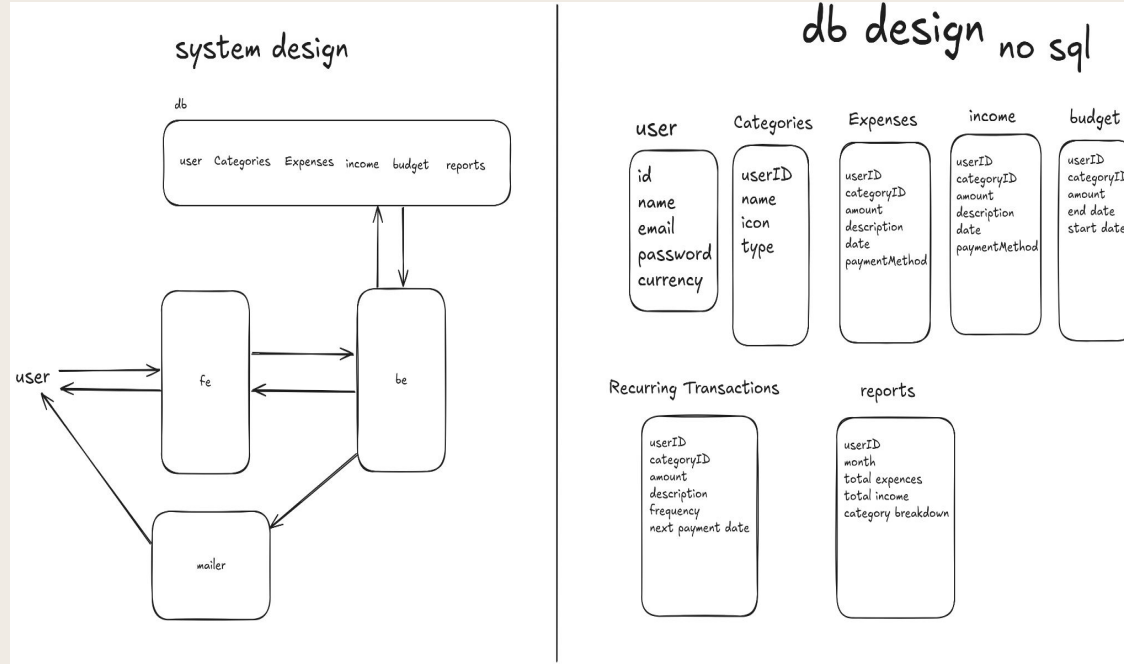
Database

System Design and MongoDB



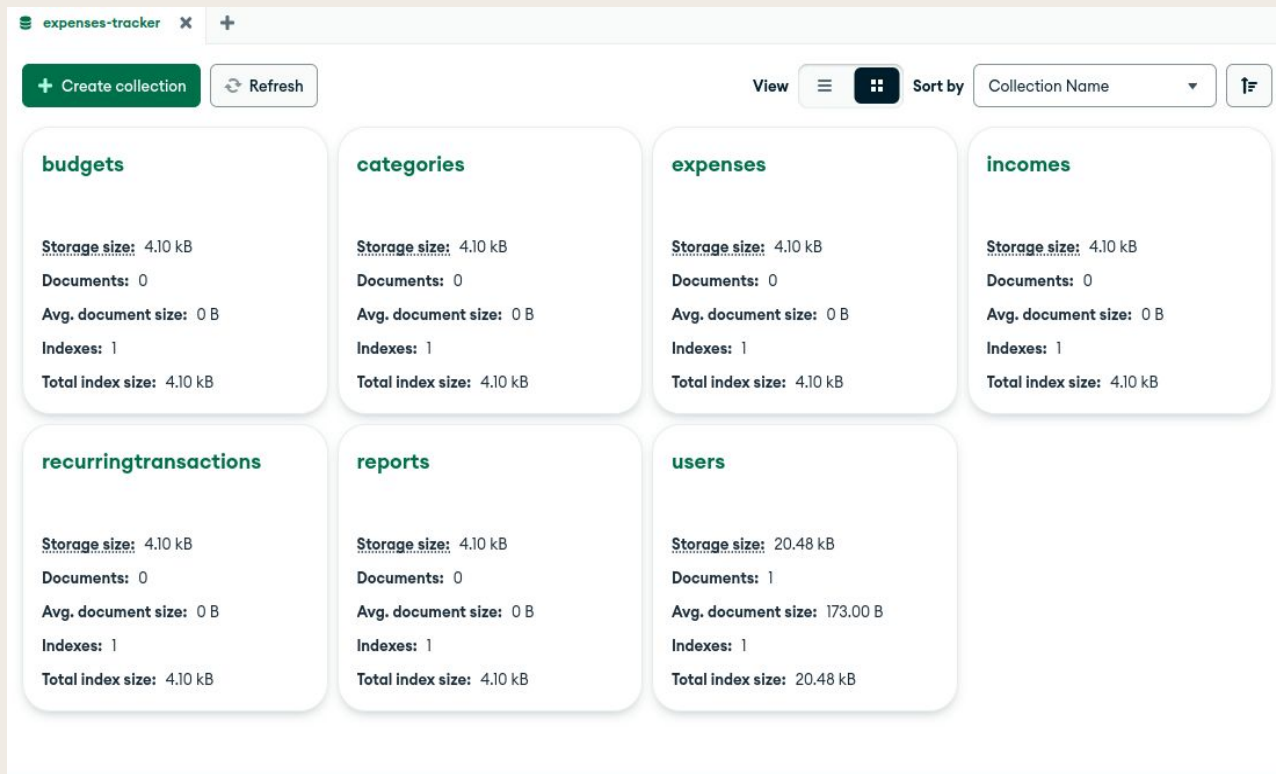
System Design

- The left side shows the **system architecture** connecting the user, frontend (fe), backend (be), and mailer service.
- The **backend** handles communication with the **database** and **mailer**, while the frontend manages user interaction.
- The right side illustrates the **NoSQL database design** using MongoDB collections.
- Key collections include:
 - **user**, **categories**, **expenses**, **income**, **budget**, **reports**, and **recurring transactions**.
- Each collection is structured to support flexibility, scalability, and clear data relationships.



MongoDB

- This view shows the MongoDB collections in the expenses-tracker database.
- Collections include: **budgets**, **categories**, **expenses**, **incomes**, **recurringtransactions**, **reports**, and **users**.
- Each collection is designed to store specific types of financial data.
- Currently, only the users collection has documents; all other collections are empty.
- The structure is ready to handle future data like expenses, incomes, and reports.



The screenshot displays the MongoDB Compass interface for the 'expenses-tracker' database. The interface includes a top navigation bar with a tab for 'expenses-tracker', a '+ Create collection' button, and a 'Refresh' button. On the right, there are controls for 'View' (list or grid), 'Sort by' (set to 'Collection Name'), and a filter icon. The main area shows seven collection cards arranged in a grid. Each card provides the following information: collection name, storage size (4.10 kB for most, 20.48 kB for 'users'), document count (0 for most, 1 for 'users'), average document size (0 B for most, 173.00 B for 'users'), index count (1 for all), and total index size (4.10 kB for most, 20.48 kB for 'users').

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
budgets	4.10 kB	0	0 B	1	4.10 kB
categories	4.10 kB	0	0 B	1	4.10 kB
expenses	4.10 kB	0	0 B	1	4.10 kB
incomes	4.10 kB	0	0 B	1	4.10 kB
recurringtransactions	4.10 kB	0	0 B	1	4.10 kB
reports	4.10 kB	0	0 B	1	4.10 kB
users	20.48 kB	1	173.00 B	1	20.48 kB



03

Backend

Files System

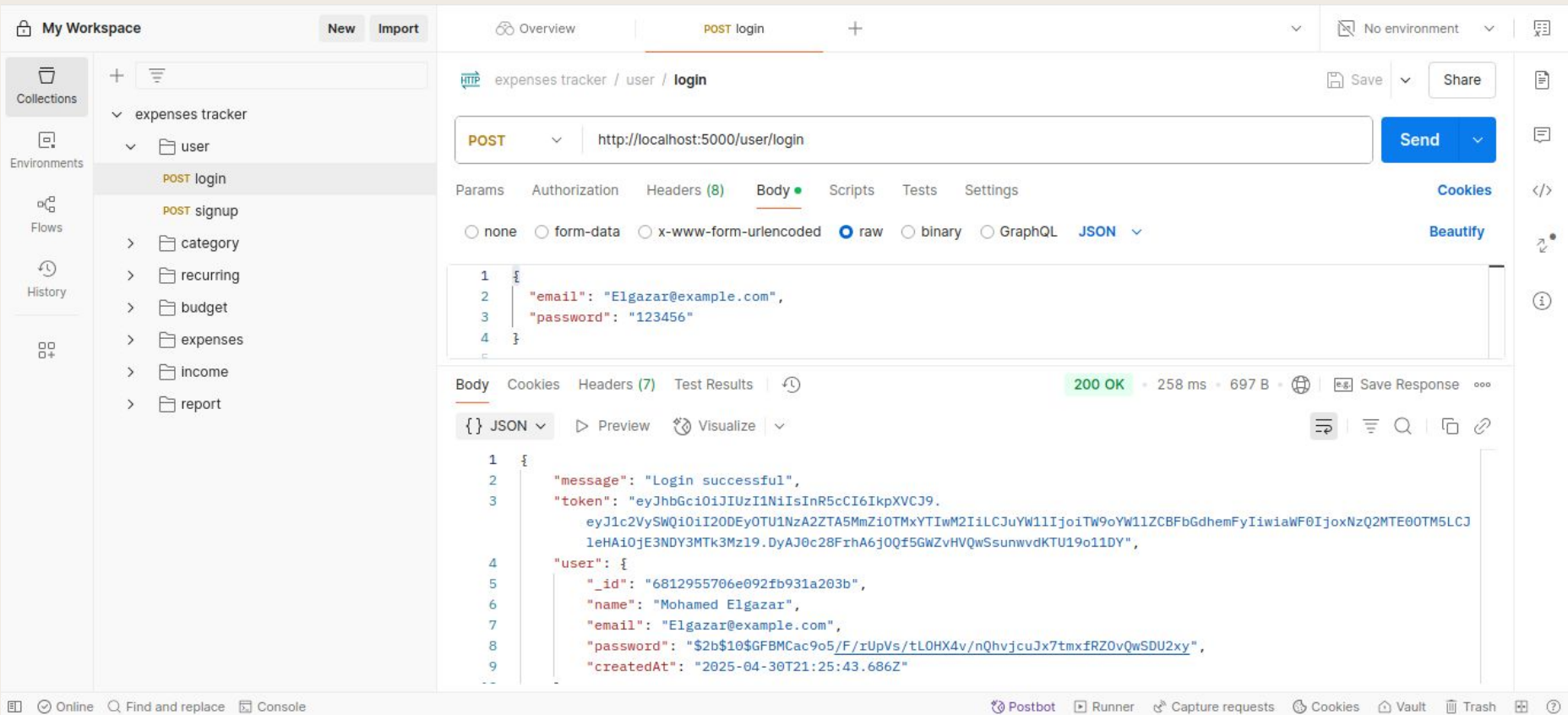


Files System

- The project is organized by feature (budget, income, expenses, etc.).
- Each feature has its own controller and routes in `src/modules`.
- Database models (schemas) are stored in the `models` folder.
- Middleware (like token verification) is in the `middleware` folder.
- `index.js` is the main entry point.
- Dependencies are managed with `package.json`.

```
Backend
├── databases
│   ├── dbConnection.js
│   └── models
│       ├── budget.model.js
│       ├── category.model.js
│       ├── expenses.model.js
│       ├── income.model.js
│       ├── recurringTransaction.model.js
│       ├── report.model.js
│       └── user.model.js
├── index.js
├── node_modules
│   └── Downloaded Modules
├── package.json
├── package-lock.json
└── src
    ├── middleware
    │   └── verifyToken.js
    └── modules
        ├── budget
        │   ├── budget.controller.js
        │   └── budget.routes.js
        ├── categories
        │   ├── category.controller.js
        │   └── category.routes.js
        ├── expenses
        │   ├── expenses.controller.js
        │   └── expenses.routes.js
        ├── income
        │   ├── income.controller.js
        │   └── income.routes.js
        ├── recurringTransaction
        │   ├── recurringTransaction.controller.js
        │   └── recurringTransaction.routes.js
        ├── reports
        │   ├── reports.controller.js
        │   └── reports.routes.js
        └── user
            ├── user.controller.js
            └── user.routes.js
```

Postman





04

Frontend

Coming Soon!!!!



Thanks!

