

Full Stack Development with MERN

Project Documentation

1. Introduction

- **Project Title:** Personal Expense Tracker
- **Team Members:**

1. Harsh Burman : Backend Developer
2. Ayush Pandey : Frontend Developer
3. Adamy Gupta : UI/UX
4. Yuktika Mishra : API Testing

2. Project Overview

- **Purpose:**
The Expense Tracker app helps users manage their daily finances by tracking income and expenses, giving a clear overview of their spending habits over time.
- **Features:**
 - User Authentication (Sign Up / Log In)
 - Add, Edit, Delete expenses and income
 - Categorize transactions
 - Real-time balance display
 - Transaction history
 - Responsive user interface

3. Architecture

- **Frontend:**
Built using **React.js** with hooks and context API for state management. Axios is used for making API calls. UI components styled with TailwindCSS or Bootstrap.
- **Backend:**
Developed with **Node.js** and **Express.js**. Follows MVC architecture. Uses RESTful API principles for communication.
- **Database:**
MongoDB is used as the database to store user data and transactions. **Mongoose** is used as an ODM for schema definition and queries.

4. Setup Instructions

- **Prerequisites:**
 - Node.js (v14 or above)
 - MongoDB (local or cloud instance e.g., MongoDB Atlas)
 - Git

- **Installation:**

```
git clone https://github.com/name/expense-tracker.git
cd Expense Tracker
```

- **Setup frontend:**

```
cd client
npm install
```

- **Setup backend:**

```
cd ../server
npm install
```

- **Add .env in /server with:**

```
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret
```

5. Folder Structure

- **Client (React):**

```
client/
├── public/
├── src/
│   ├── components/
│   ├── pages/
│   ├── context/
│   ├── App.js
│   └── index.js
└── package.json
```

- **Server (Node.js + Express):**

```
server/
├── controllers/
├── models/
├── routes/
├── middleware/
├── config/
├── server.js
└── package.json
```

6. Running the Application

- **Frontend:**

```
cd client
npm start
```

- **Backend:**

```
cd server
npm start
```

7. API Documentation

Endpoint	Method	Description
/api/auth/register	POST	Register new user
/api/auth/login	POST	Authenticate user
/api/transactions/	GET	Get all user transactions
/api/transactions/	POST	Add a new transaction
/api/transactions/:id	PUT	Update transaction
/api/transactions/:id	DELETE	Delete transaction

- Example Response:

```
{
  "transactions": [
    {
      "_id": "abc123",
      "amount": -500,
      "category": "Food",
      "date": "2025-04-10T12:00:00Z"
    }
  ]
}
```

8. Authentication

- Uses **JWT (JSON Web Tokens)** for authentication.
- Tokens are generated on login and stored in the browser (localStorage).
- Protected routes and middleware used for secure access to user-specific data.

9. User Interface

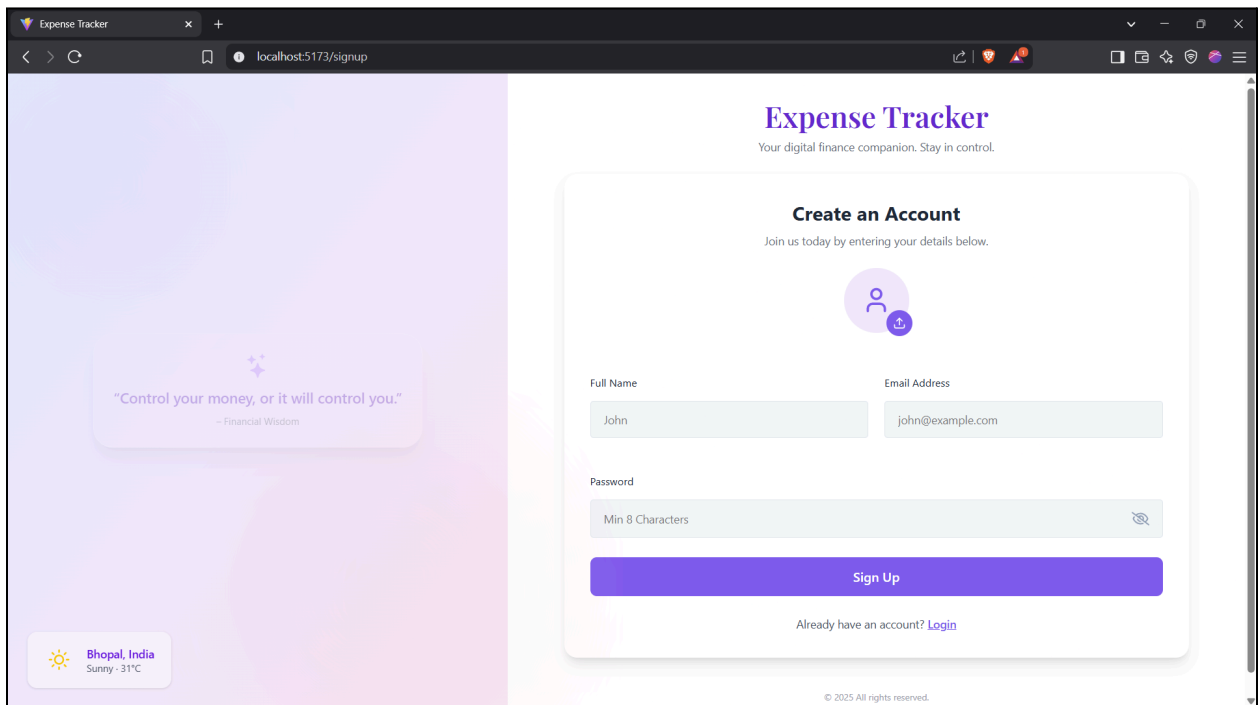
- Responsive layout
- Dashboard with balance chart
- Simple forms for adding transactions

10. Testing

- **Frontend:** Jest + React Testing Library
- **Backend:** Postman for API testing, Jest/Mocha for unit testing (optional)

11. Screenshots or Demo

- **Sign up**




The screenshot shows a web browser window with the title "Expense Tracker" and the URL "localhost:5173/signup". The page features a light purple and pink background. On the left, there is a quote: "Control your money, or it will control you." attributed to "Financial Wisdom". Below the quote is a weather widget for "Bhopal, India" showing "Sunny - 31°C". On the right, there is a "Create an Account" form. The form includes a sub-header "Join us today by entering your details below." and a user icon. The form fields are: "Full Name" (containing "John"), "Email Address" (containing "john@example.com"), and "Password" (containing "Min 8 Characters" and a toggle icon). A blue "Sign Up" button is at the bottom of the form. Below the button is a link: "Already have an account? [Login](#)". At the bottom right of the page, there is a small copyright notice: "© 2025 All rights reserved."

Expense Tracker

Your digital finance companion. Stay in control.

Create an Account

Join us today by entering your details below.



Full Name

John

Email Address

john@example.com

Password

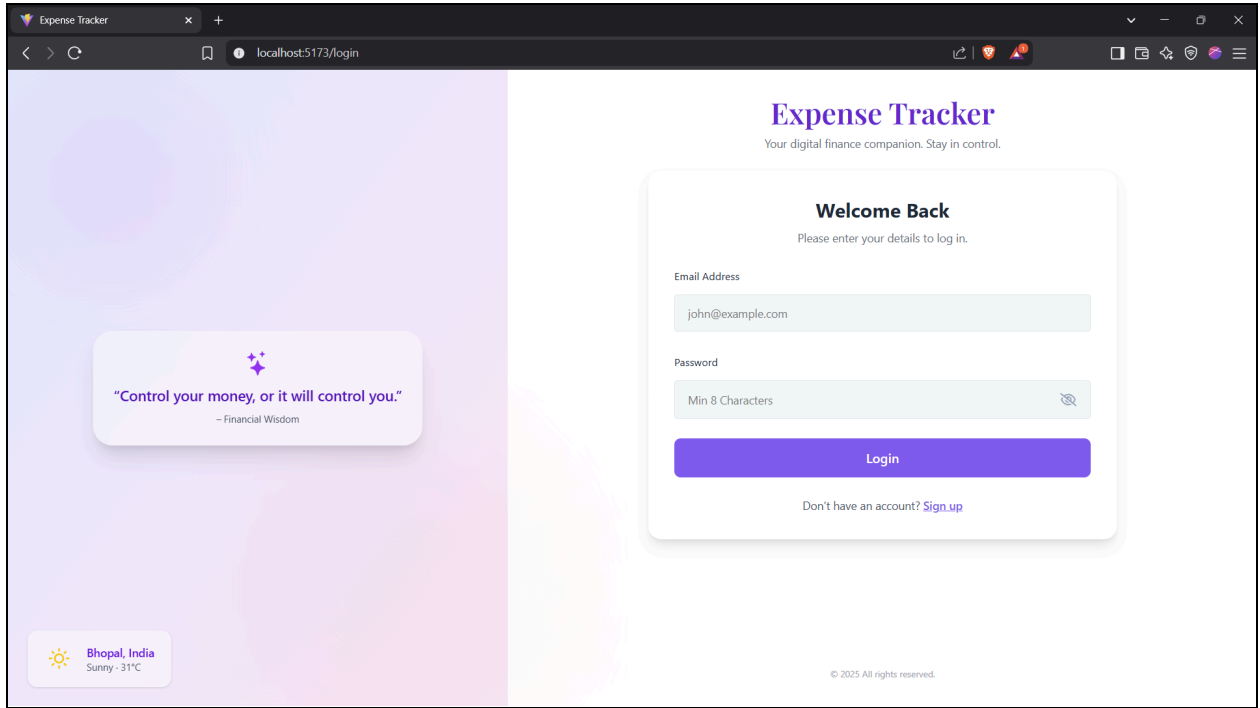
Min 8 Characters

[Sign Up](#)

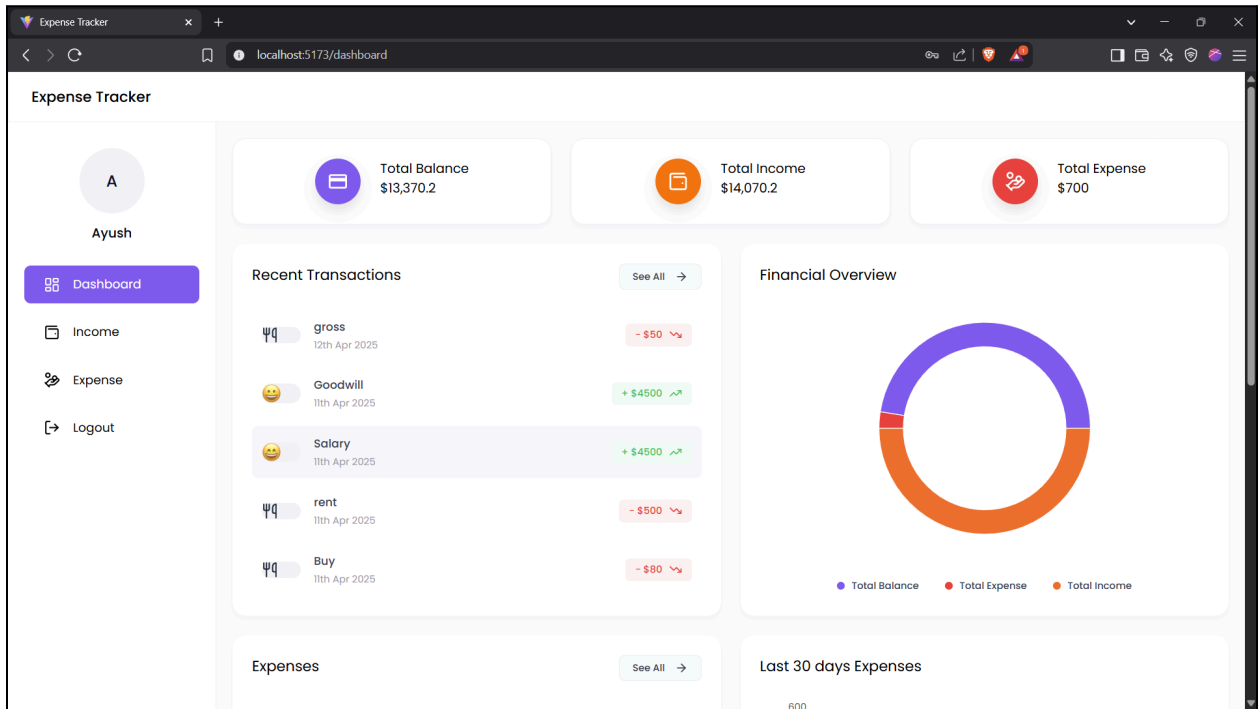
Already have an account? [Login](#)

© 2025 All rights reserved.

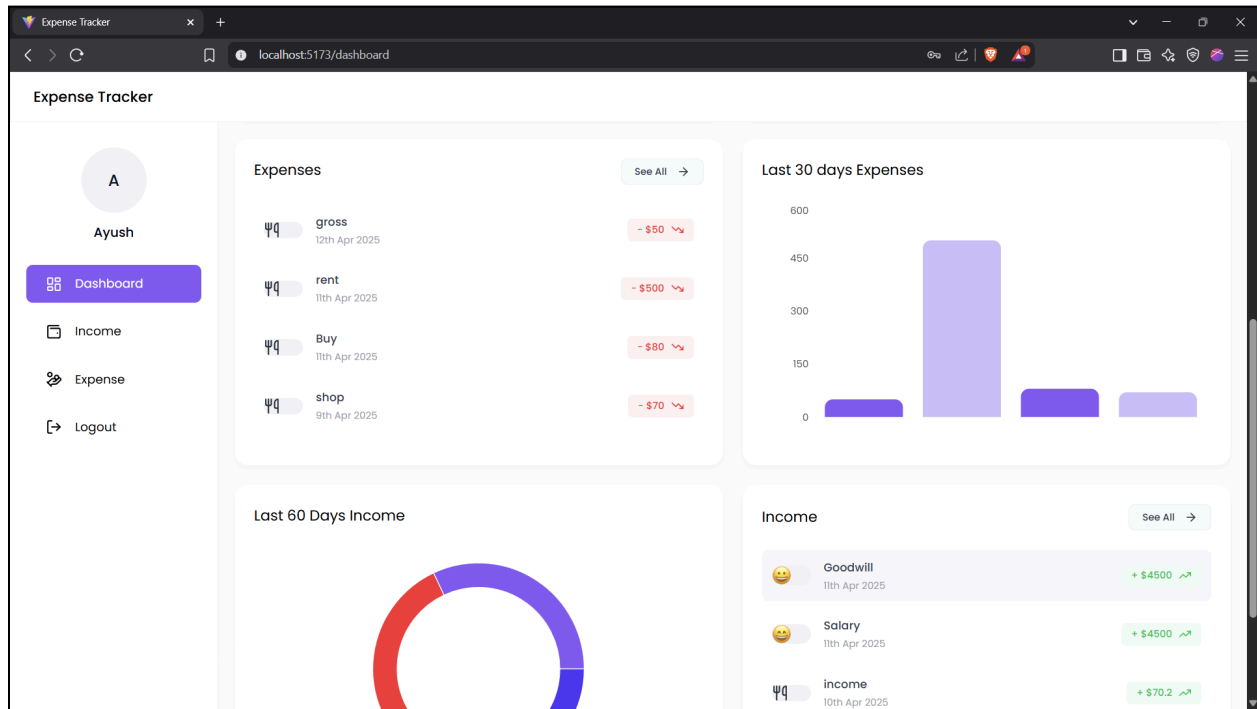
- **Login**



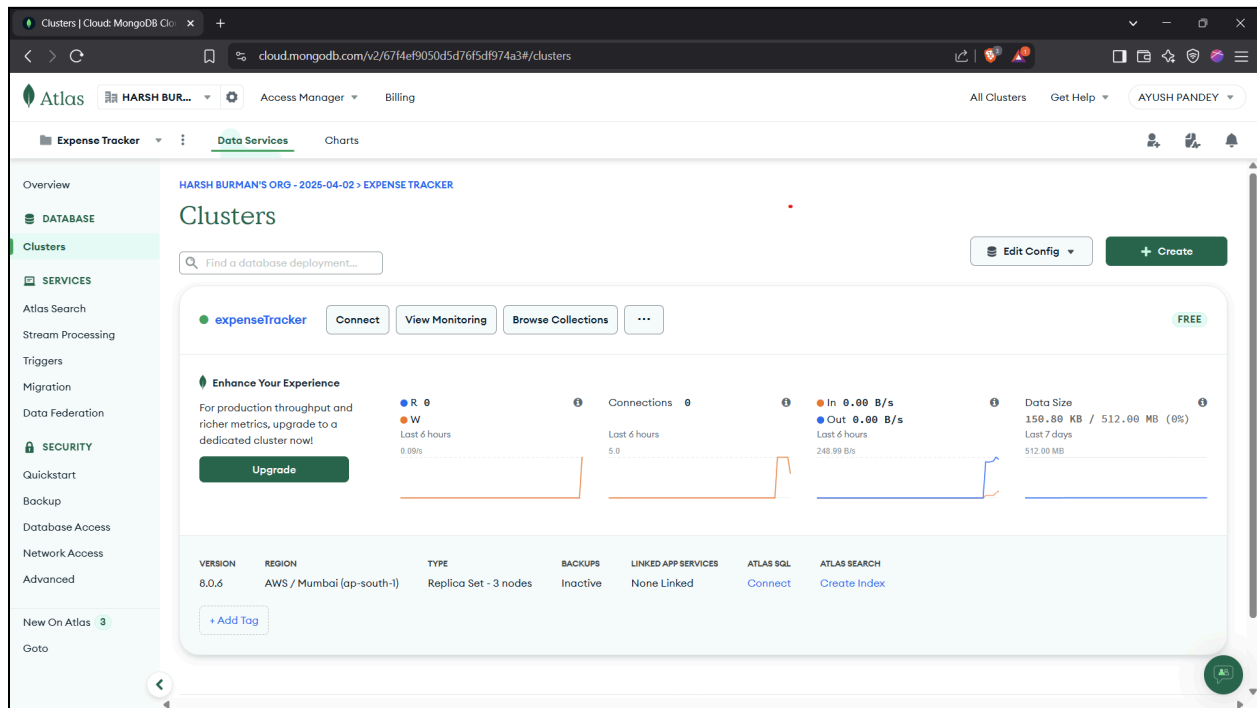
• Dashboard1



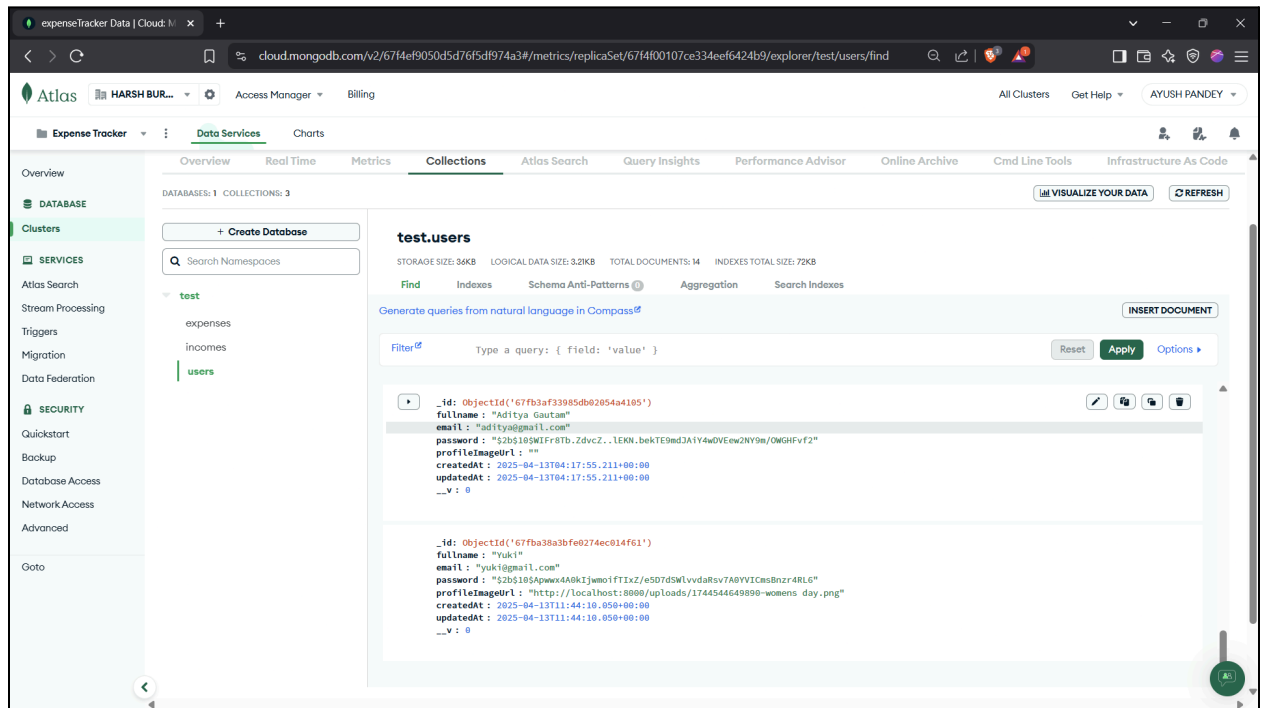
• Dashboard2



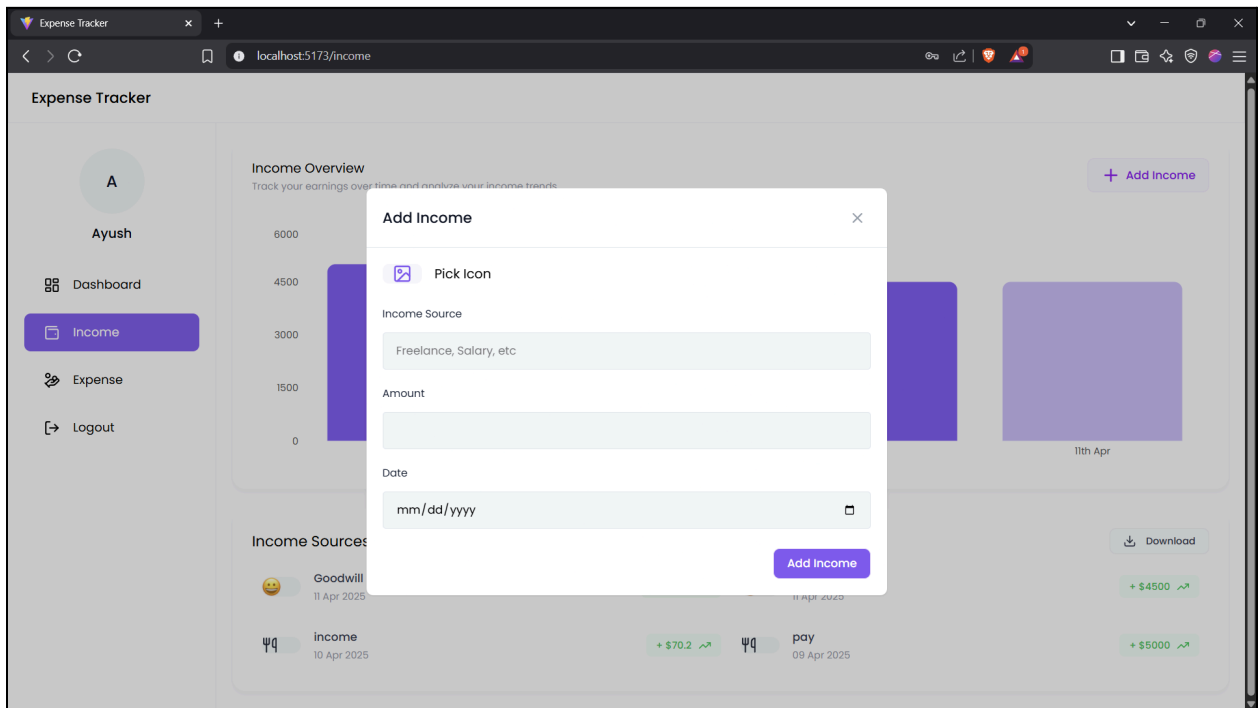
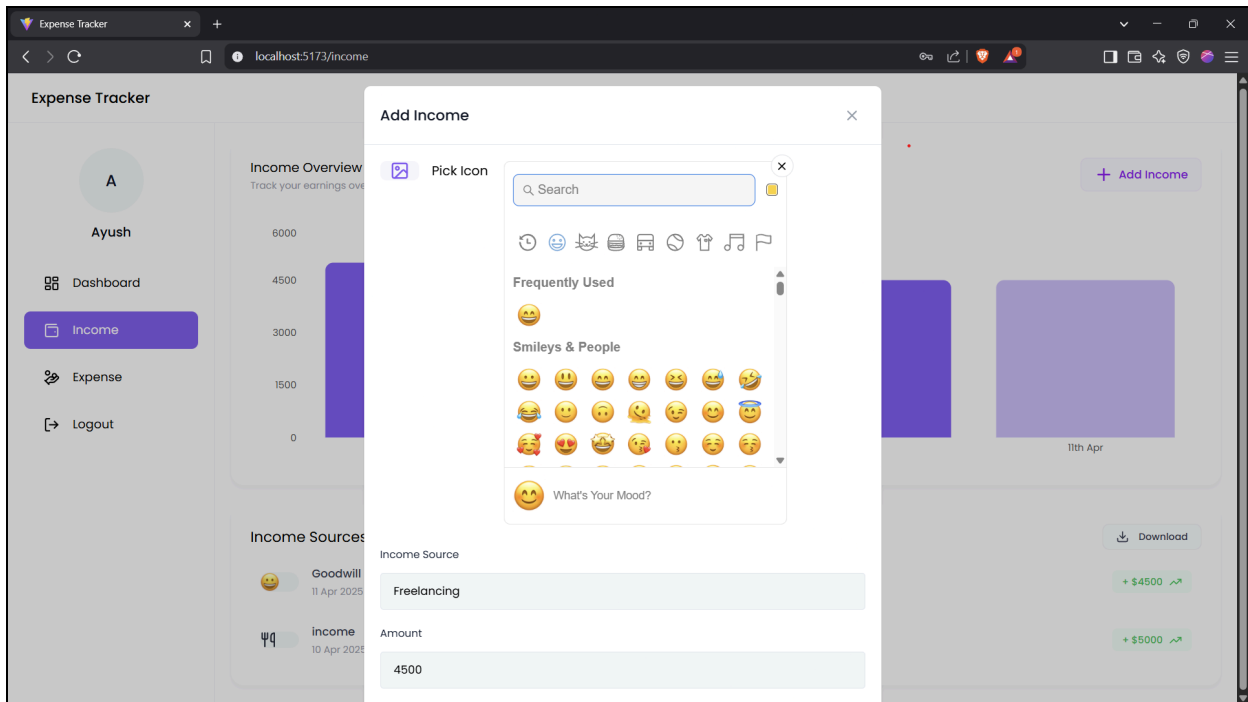
• Database Cluster



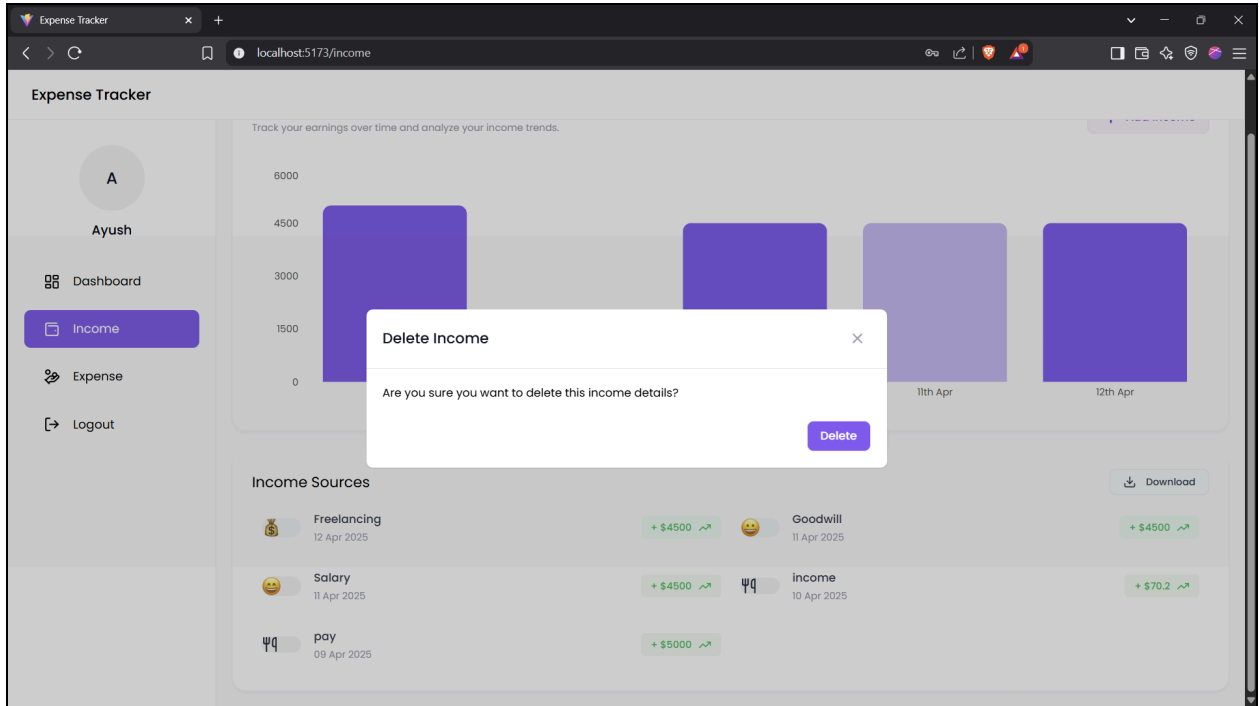
- Database Users



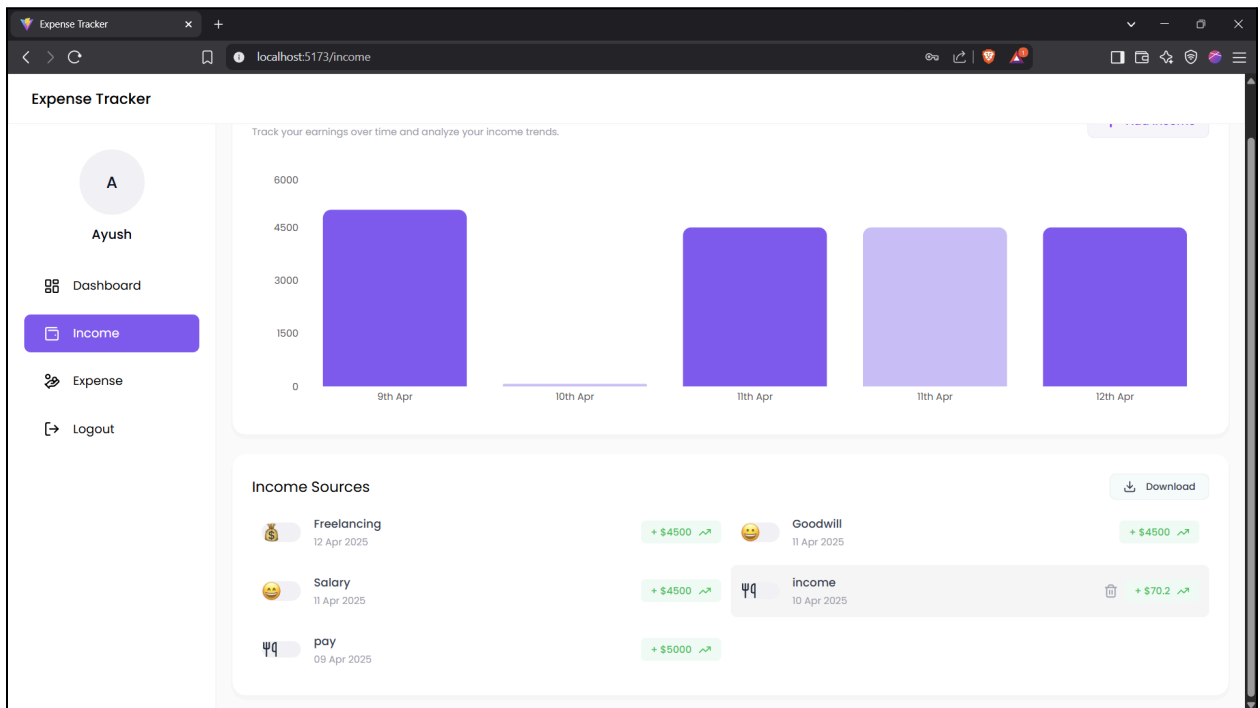
- Add Income:



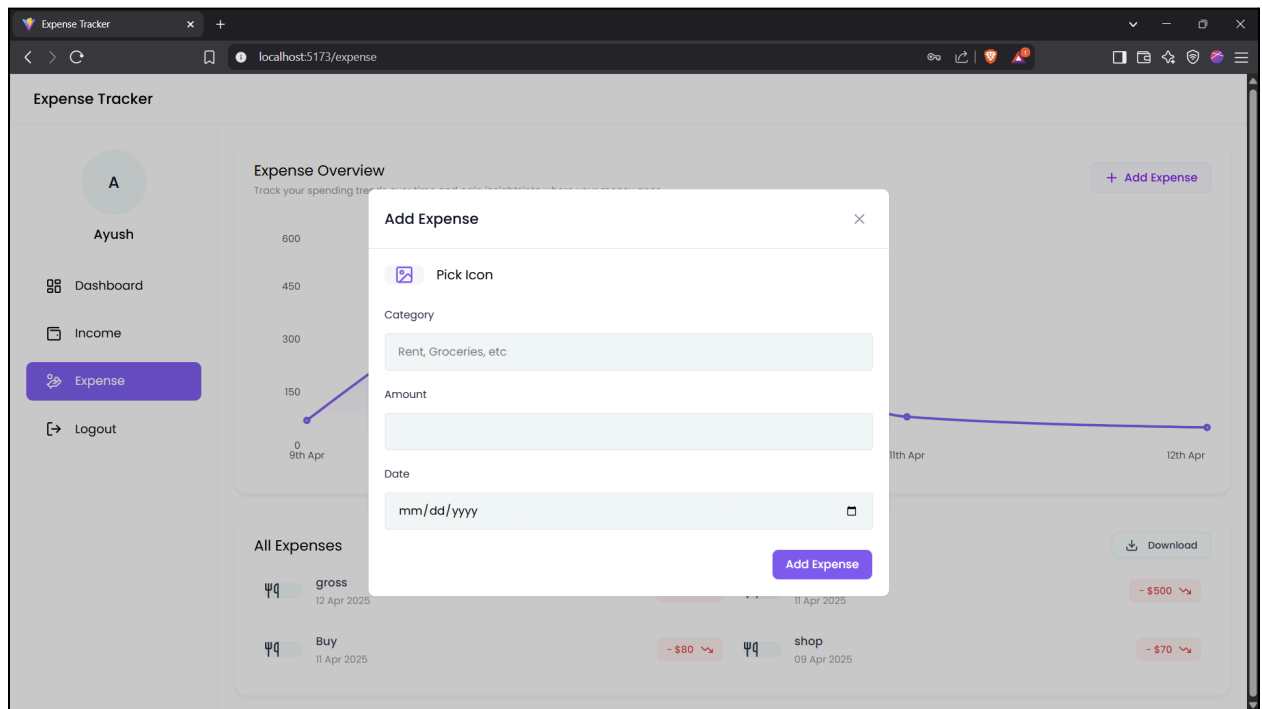
- **Delete Income:**



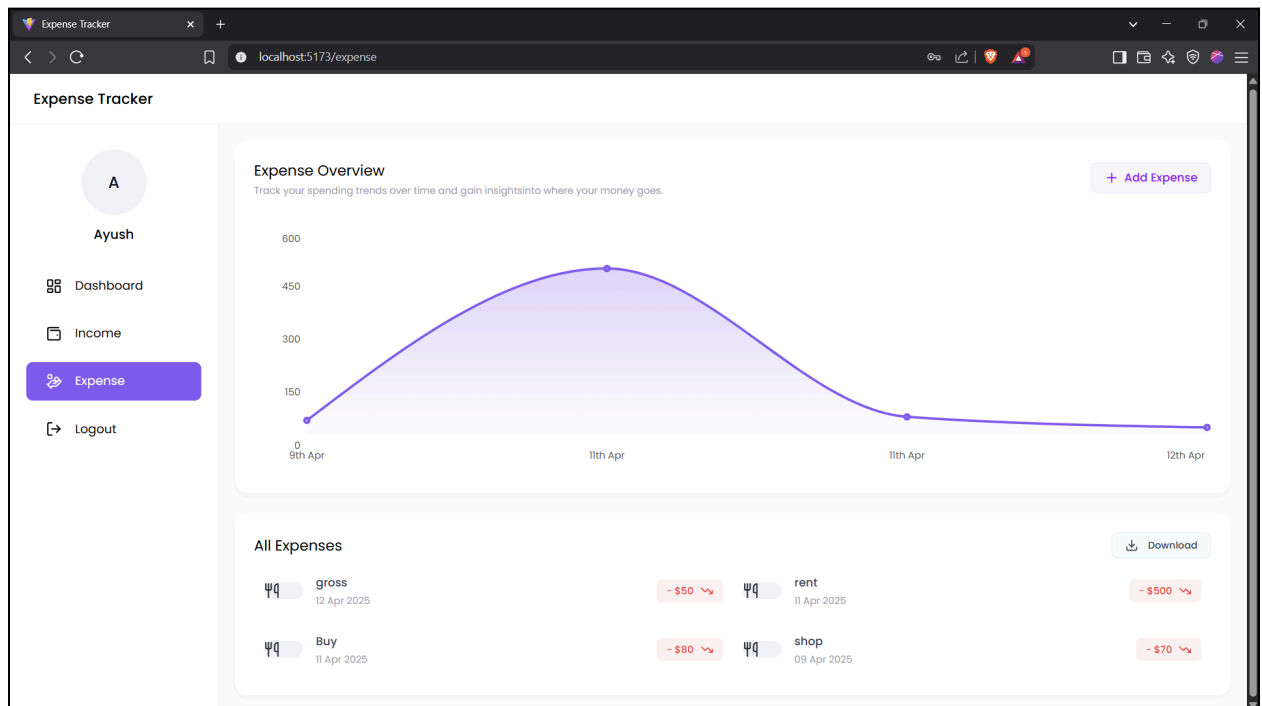
- Income Sources



- Add Expense



- Expense Tracking Graph



12. Known Issues

- No multi-user budgeting support
- Needs input validation on frontend
- Error messages not shown clearly in some component

13. Future Enhancements

- Monthly report and charts with D3.js or Chart.js
- Budgeting goals and alerts
- Export data to Excel/PDF
- Add support for recurring transactions
- Mobile app version using React Native