

Expense-Tracker

Course: Backend Development

Lecturer: Hok Tin

Data Science G1

Team 8:

- Chum Phalla
- Khy Pichsereyvatanak
- Soeun Sokchetra

Date: 27th July 2025

Group : DS G1



Team Member



Chum Phalla



Khy Pichsereyvatanak



Soeun Sokchatra

Task Division

Name	Chum Phalla	Khy Pichsereyvathanak	Soeun Sokchatra
Frontend	Sidebar Navigation for manage expense & income	Authentication pages	Dashboard UI Page
Backend	<ul style="list-style-type: none">• Set up Server.js and routers• Define all API endpoints and connect them to controllers.• Implement middleware• Ensure proper request validation and response formatting.	<ul style="list-style-type: none">• Implement controller functions• Handle request processing, business rules, and data transformation.• Integrate authentication logic (JWT, bcrypt).	<ul style="list-style-type: none">• Design and implement Sequelize models• Set up database connection and configuration.• Handle associations, and data validation• Seed initial data and write utility scripts for DB management.

Why expense-Tracker?

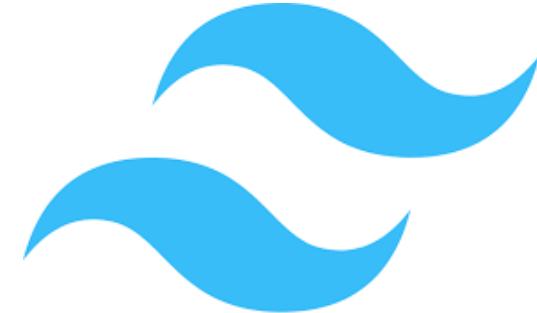
We chose the Expense Tracker project because it solves a real-world problem and provides a complete full-stack development experience. It involves essential features like user **authentication** with JWT, **CRUD** operations, and date-based **filtering**. The project is practical, easy to understand, and highly **extendable**, making it a great opportunity to apply and reinforce key concepts in **frontend**, **backend**, and **database** integration.



Technology Used



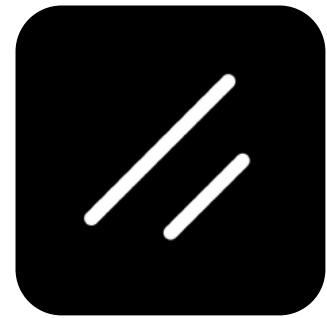
Next.js 14



Tailwind CSS



React Hook Form



shadcn/ui



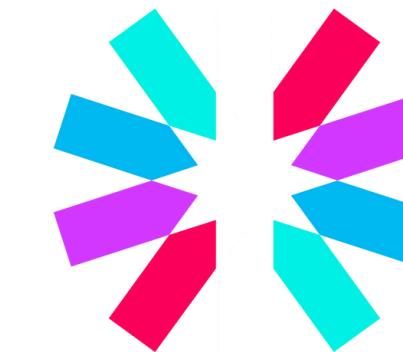
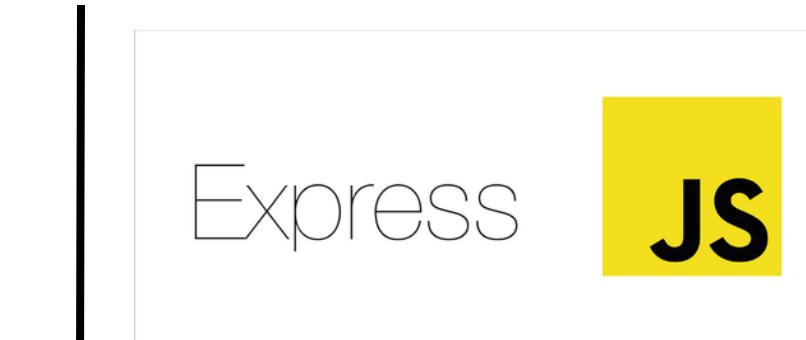
TypeScript



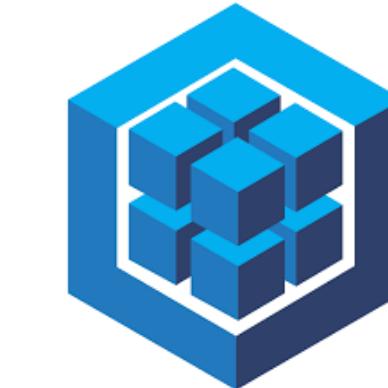
date-fns



Lucide React

JWT
Authentication

bcryptjs



Sequelize ORM

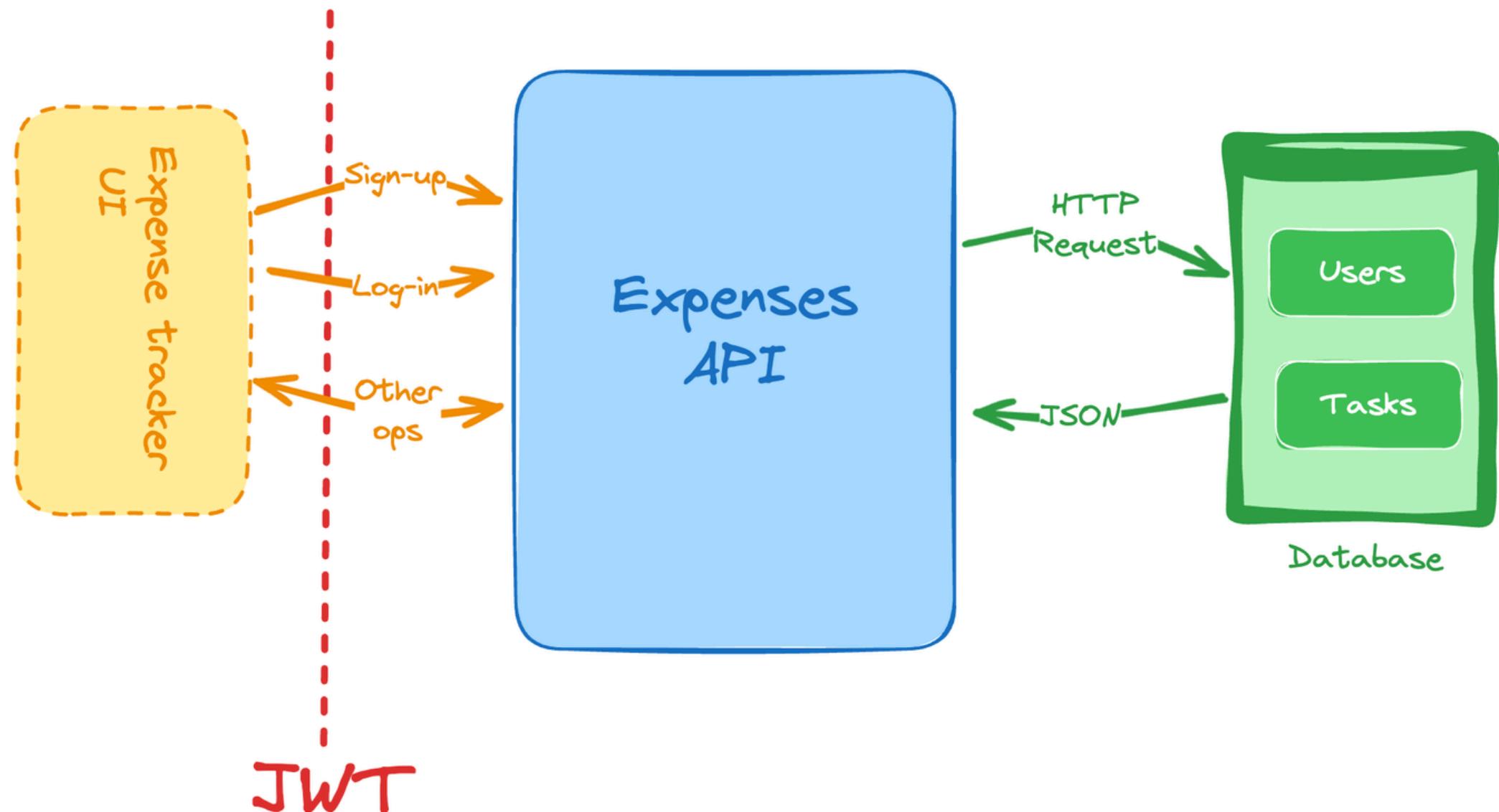


Frontend

Backend

page | 2

System Architecture



After users **log in**, the system gives them a secure token (**JWT**) to prove who they are. This token is used to access protected features without needing to log in again each time.

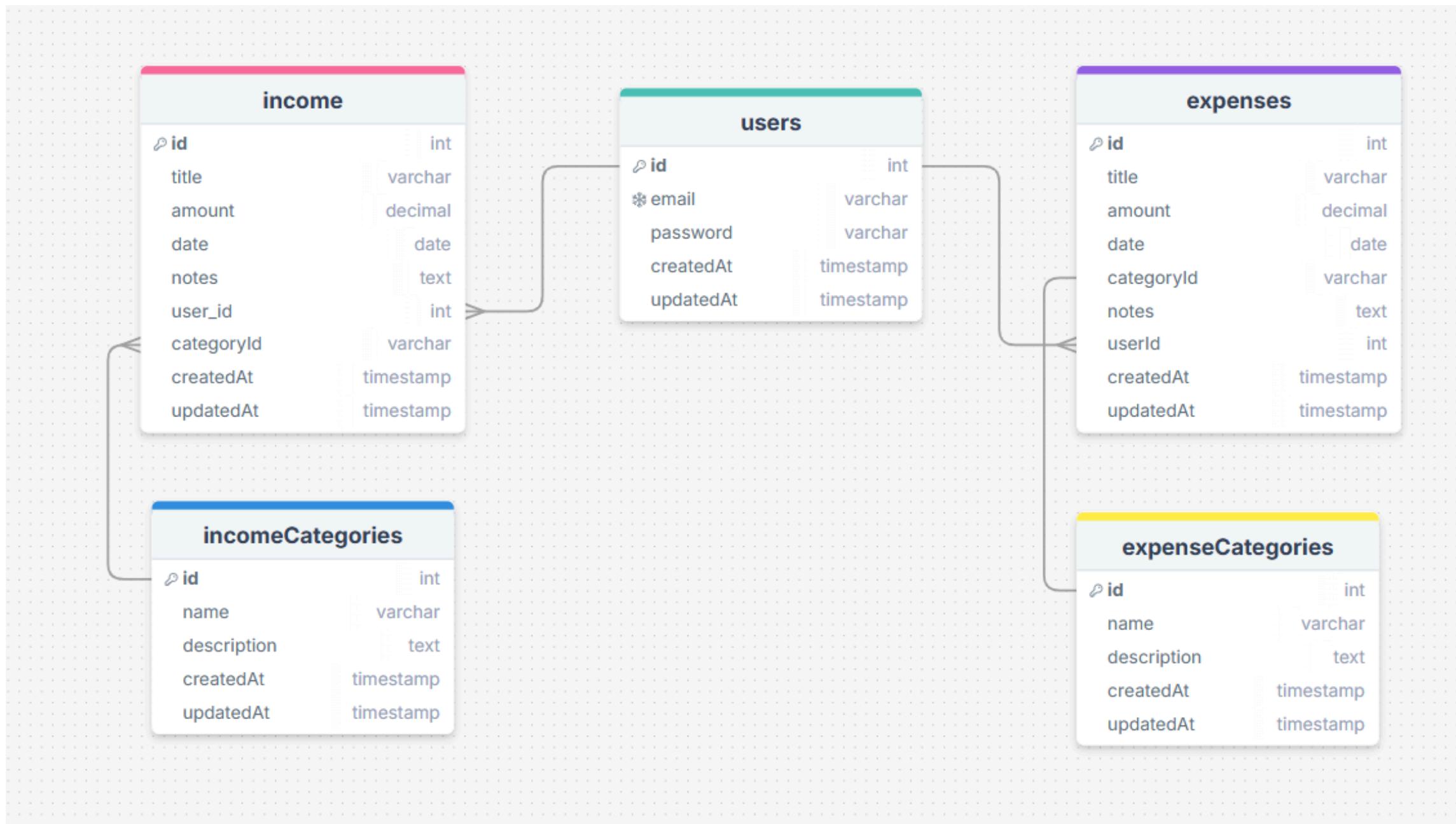
Key Features

 Authentication	 Expense & Income Management	 Dashboard & Analytics	 User Interface
<i>User registration with email and password</i>	<i>Create new entries with title, amount, date, category, and notes</i>	<i>Summary cards for weekly, monthly, and 3-month totals</i>	<i>Clean, modern UI with green primary theme</i>
<i>Secure login using JWT tokens</i>	<i>Update or remove entries with confirmation</i>	<i>Time-based filters: week, month, 3 months, custom range</i>	<i>Supports dark mode toggle</i>
<i>Forgot password & reset support</i>	<i>Each user has their own set of income and expense categories</i>	<i>Date range picker for detailed filtering</i>	<i>Fully responsive across all devices</i>
<i>Session/token management for secure access</i>	<i>Create, view, and manage categories filtered by logged-in user</i>	<i>Color-coded categories for easy viewing</i>	<i>Toast alerts for actions and errors</i>

Project Structure

```
Expense_Tracker/
├── backend/                      # Express.js backend
│   ├── controllers/              # Route controllers (user, expense, income, etc.)
│   ├── db/                       # Database config and seeders
│   ├── models/                   # Sequelize models
│   ├── routes/                   # API route definitions
│   └── server.js                 # Express app entry point
│       ...
└── frontend/                    # Next.js 14 frontend
    ├── app/                      # App Router pages
    ├── components/               # Reusable React components
    ├── lib/                       # Utility functions
    ├── public/                    # Static assets
    ├── styles/                   # Global styles (if any)
    └── ...
    ...
└── README.md                     # Project documentation
```

Relational Schema



a **User** has many **Income** and **Expense** records, while each **Income** belongs to a **User** and each **Expense** belongs to a **User**. An **IncomeCategory** has many **Income** records, and each **Income** belongs to an **IncomeCategory**. Similarly, an **ExpenseCategory** has many **Expense** records, and each **Expense** belongs to an **ExpenseCategory**.

API Endpoint

Category	Method	Endpoint	Description
 Authentication	POST	/api/users/register	Register a new user
	POST	/api/users/login	Login, returns JWT token + user info
	POST	/api/users/forgot-password	Request password reset
 Expense Management	GET	/api/expenses	Get all expenses (supports filters)
	POST	/api/expenses	Add a new expense
	GET	/api/expenses/:id	Get expense by ID
	PUT	/api/expenses/:id	Update an existing expense
	DELETE	/api/expenses/:id	Delete an expense
 Summary & Stats	GET	/api/expenses/summary/stats	Get total expenses for week/month/3 mo.

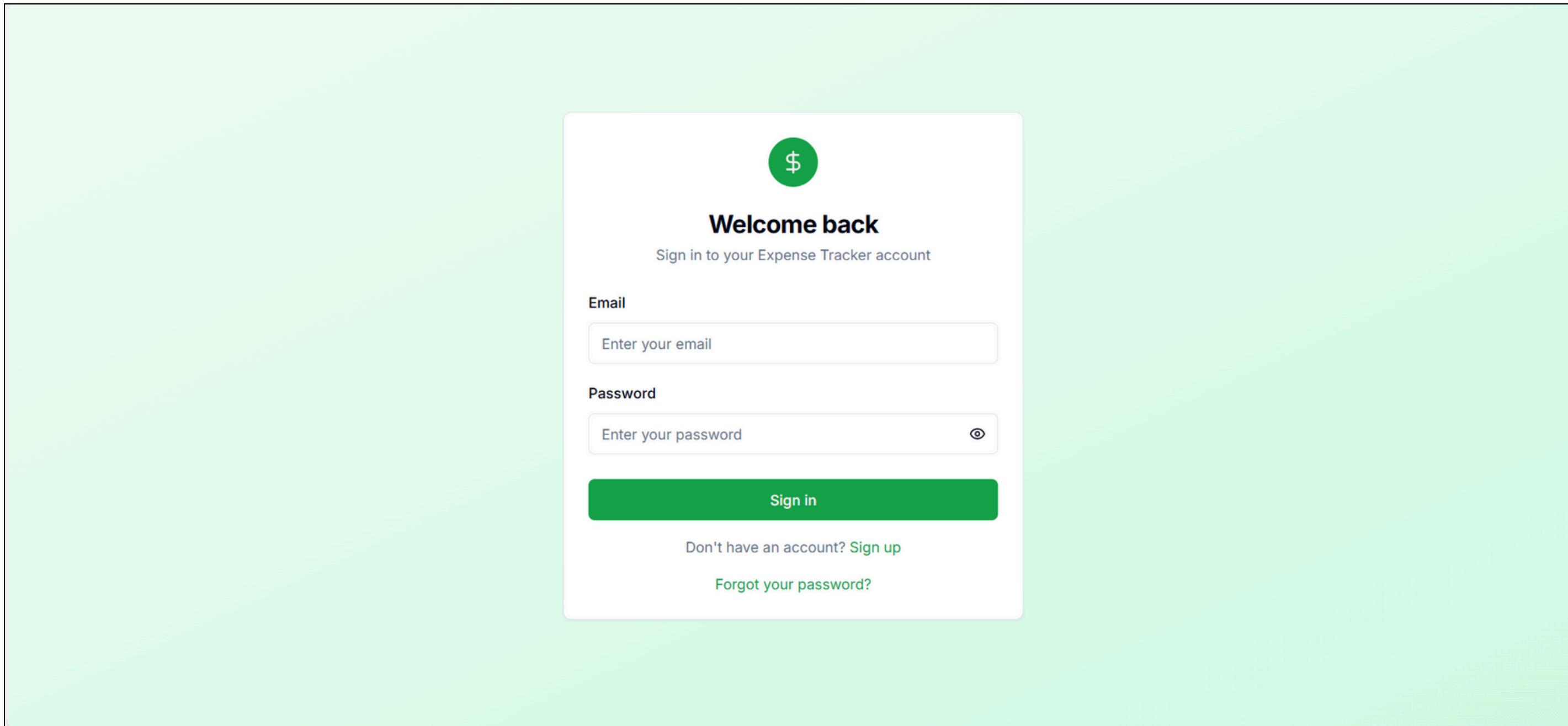
Category	Method	Endpoint	Description
 income management	GET	/api/incomes	get all income
	POST	/api/incomes	create new income
	GET	/api/incomes/id	get income by specific id
	PUT	/api/incomes/id	update on specific income
	DELETE	/api/incomes/id	delete income

Category	Method	Endpoint	Description
 Category for expense	GET	/api/categories	get all category
	POST	/api/categories	create new category
	GET	/api/categories/:id	get category by specific id
	PUT	/api/categories/:id	update on specific category
	DELETE	/api/categories/:id	delete category
 Category for income	GET	/api/income-categories	get all category
	POST	/api/income-categories	create new category
	GET	/api/income-categories/:id	get category by specific id
	PUT	/api/income-categories/:id	update on specific category
	DELETE	/api/income-categories/:id	delete category

Seeding Logic

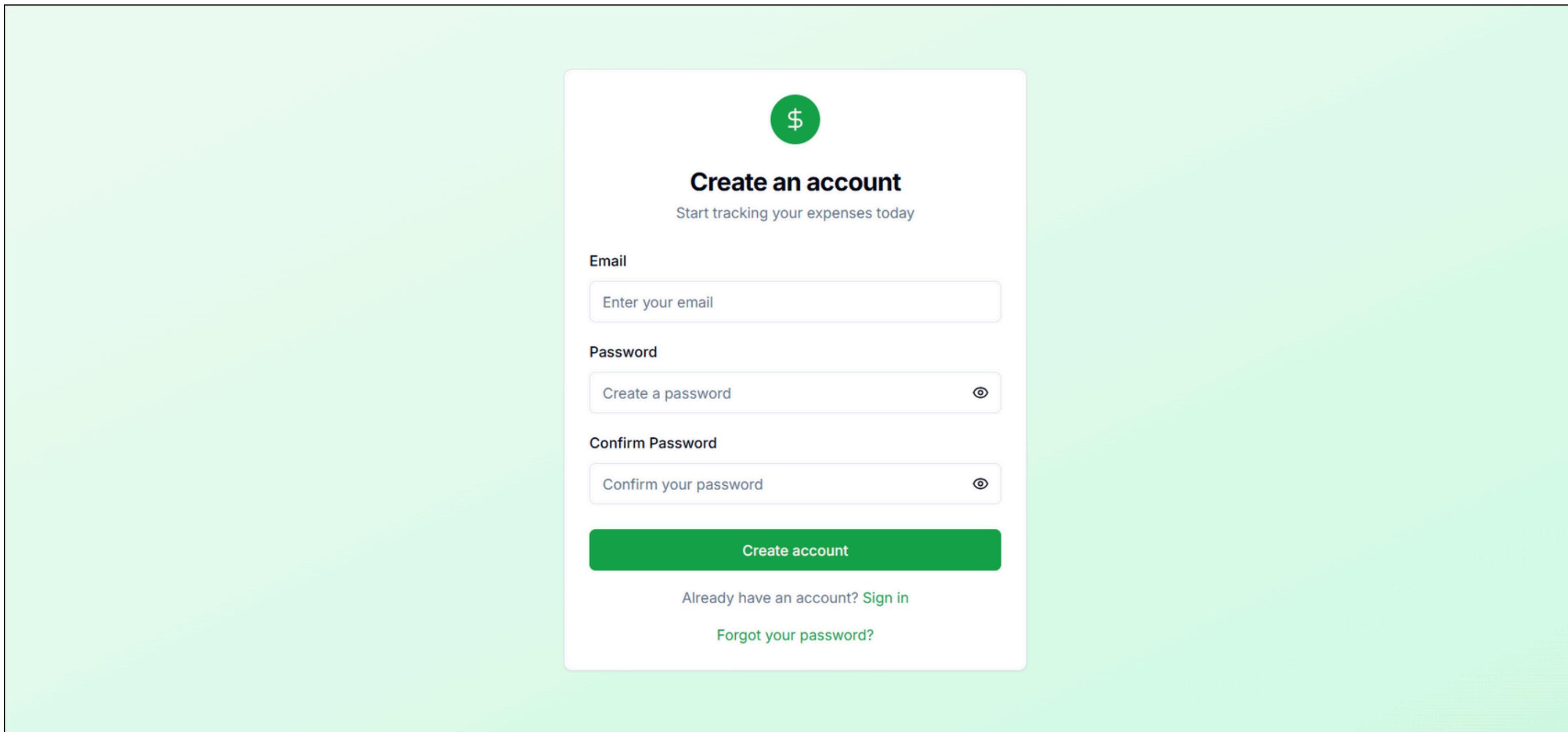
- **Seed runs only once:** On first run, if no users exist, initial users and categories are created.
- **User-specific categories:** Each user gets their own set of income and expense categories.
- **Passwords are hashed before saving users.**

User interface



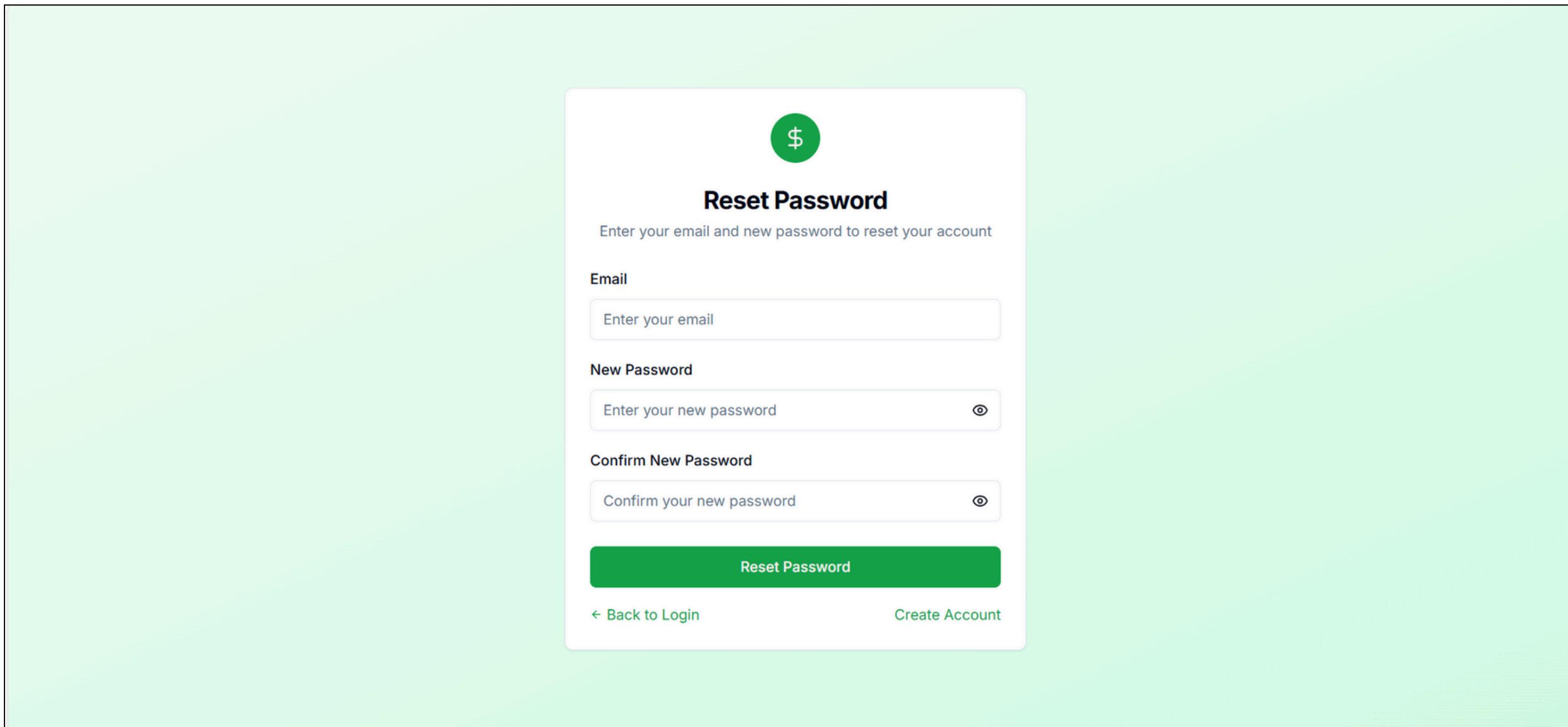
login page

User interface



sign up page

User interface



forgot-password page

User interface

The screenshot shows the 'Expense Tracker' application's dashboard. On the left is a sidebar with a green header containing a dollar sign icon and the text 'Expense Tracker' and 'Manage your finances'. Below this are navigation links: 'Dashboard' (highlighted in green), '+ Add Expense', 'Add Income', 'Manage Incomes', 'Expense Categories', 'Income Categories', an email link 'vathanak@gmail.com', and a 'Logout' button. The main content area has a title 'Dashboard' and a subtitle 'Track and manage your expenses efficiently'. It displays financial summary statistics: 'Total Income: \$1800.00', 'Total Expense: \$95.00', and 'Balance: \$1705.00'. Below these are three cards: 'This Week' (\$0.00, Avg: \$0.00/day), 'This Month' (\$95.00, Avg: \$23.75/week), and 'Last 3 Months' (\$95.00, Avg: \$31.67/mo). A section titled 'Recent Expenses' allows filtering by time ('All Time') and category ('All'). A specific expense entry is shown: 'Bus Pass' under 'Transport' on 'Jul 4, 2025' for '\$15.00'. There are edit and delete icons next to the expense entry.

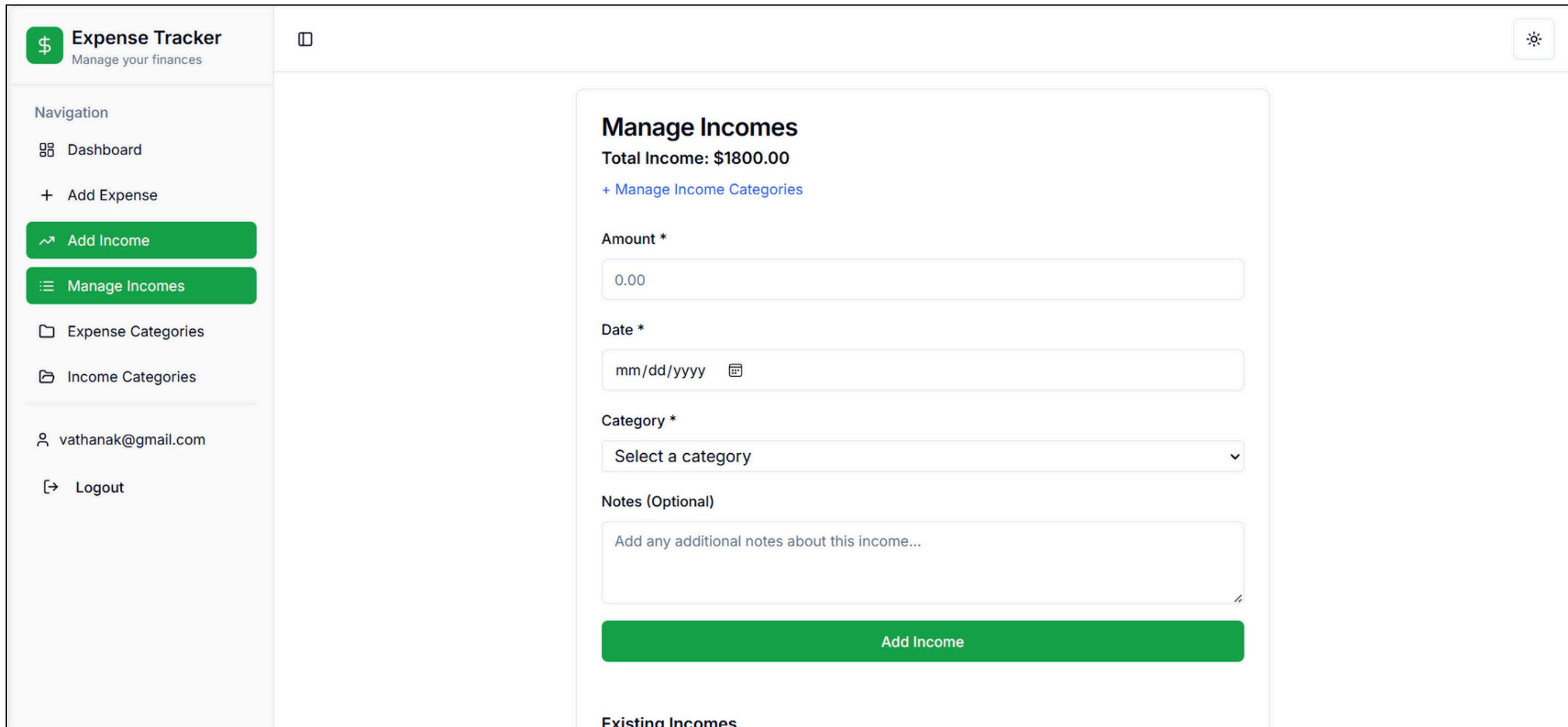
dashboard page

User interface

The screenshot shows the 'Expense Tracker' application interface. On the left is a sidebar with navigation links: 'Dashboard' (highlighted in green), '+ Add Expense' (button), 'Add Income', 'Manage Incomes', 'Expense Categories', 'Income Categories', 'vathanak@gmail.com', and 'Logout'. The main area has a title 'Add New Expense' and a subtitle 'Enter the details of your new expense'. It contains fields for 'Title *' (placeholder 'e.g., Grocery Shopping'), 'Amount *' (placeholder '0.00'), 'Date *' (placeholder 'mm/dd/yyyy'), 'Category *' (dropdown menu 'Select a category' with a '+ Add new category' link), and a 'Notes (Optional)' text area ('Add any additional notes about this expense...'). At the bottom are 'Add Expense' and 'Cancel' buttons.

Add Expense page

User interface



Add&manage Income page

User interface

The screenshot shows the 'Expense Tracker' application interface. On the left is a sidebar with the following navigation items:

- Dashboard
- + Add Expense
- ↗ Add Income
- :≡ Manage Incomes
- Expense Categories** (highlighted in green)
- Income Categories

Below the sidebar, the user information is displayed: vathanak@gmail.com and Logout.

The main content area is titled "Manage Categories". It contains fields for "Name *" (with placeholder "e.g., Groceries") and "Description (Optional)" (with placeholder "Add a description for this category..."). A large green button labeled "Add Category" is centered below these fields.

Below this section, the "Existing Categories" table is shown:

Category	Description	Edit	Delete
Entertainment	Movies, games, fun	Edit	Delete
Food	Meals and groceries	Edit	Delete
Shopping	Clothes and accessories	Edit	Delete
Transport	Commute and travel	Edit	Delete

Expenses Category page

User interface

The screenshot shows the 'Expense Tracker' application interface. On the left, there is a sidebar with the following navigation items:

- Dashboard
- + Add Expense
- ↗ Add Income
- ☰ Manage Incomes
- 📁 Expense Categories
- 📁 Income Categories (highlighted in green)
- vathanak@gmail.com
- [Logout]

The main content area is titled 'Manage Income Categories'. It contains fields for 'Name *' (with placeholder 'e.g., Salary') and 'Description (Optional)' (with placeholder 'Add a description for this category...'). A large green button labeled 'Add Category' is centered below these fields. Below this section, there is a table titled 'Existing Income Categories' with the following data:

Name	Description	Edit	Delete
Freelancing	Freelance	Edit	Delete
Gift	Received gifts	Edit	Delete
Investments	Stock/crypto earnings	Edit	Delete
Salary	Monthly salary	Edit	Delete

Income Category page

Challenges & Lesson Learned

- **Database Relationships:** Designing user-specific categories required careful model associations.
- **Authentication:** Implementing JWT and password hashing improved security but added complexity.
- **Data Seeding:** Ensuring seed data runs only once and matches user/category relationships.
- **Frontend-Backend Integration:** Handling API errors and state management for a smooth user experience.
- **Lesson Learned:** Planning data models and API endpoints early is crucial for scalable development.

Conclusion

In the end of this project, we have:

- Developed a full-stack **Expense Tracker** with Next.js and Express.js.
- Implemented secure **authentication**, **user-specific categories**, and a **modern**, responsive **UI**.
- Achieved seamless integration between **frontend** and **backend** for real-time **expense management**.

Future Work

- Add recurring expenses/incomes and **reminders**.
- Implement advanced **analytics** and export **features**.
- Enhance **mobile experience** and **accessibility**.
- Integrate **third-party** authentication (Google, Facebook).
- **Deploy** to cloud and enable **multi-user collaboration**.

References

[Backend Project Ideas](#)[Expense-Tracker API](#)

Thanks you



Happy me Happy expense tracking!