International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Design and Implementation of a Family Expense Tracker Using MERN Stack

Mukka Bhavani[1], K.Jayasri [2]

[1]Student, School of Informatics, Department of MCA, Aurora Deemed University, Hyderabad
[2]Assistant Professor, School of Engineering, Department of CSE, Aurora Deemed University, Hyderabad
[1]mukkabhavani1942@gmail.com, [2]jayasri@aurora.edu.in

ABSTRACT:

Handling family expenditures manually usually results in errors, opaqueness, and bad budgeting. This paper outlines the creation of a Family Expense Tracker based on the MERN (MongoDB, Express.js, React.js, Node.js) stack. The system provides a central utility for family heads and members to manage budgets, keep track of expenditures, track savings, and provide reports. Key features include role-based access, cost classification, PDF reporting, chart-based analytics, and JWT authentication. The solution offers accuracy, increases transparency, and encourages improved savings behavior.

Keywords: *Family Expense Tracker, MERN Stack, Expense Management, Budgeting, Savings, Web Application.*

I. INTRODUCTION

Financial planning is a key component of family planning, allowing families to plan money, meet day-to-day expenses, and gain long-term savings objectives. In the past, families have relied on notebooks, paper journals, or spreadsheets to track money in and money out. Though they have a simple format, they are susceptible to error so quickly, do not offer collaborative functionality, and cannot provide instant analysis. Due to the growth in complexity of family spending with the presence of more than a single source of income and diverse spending categories, such labor-intensive approaches are sure to lead to overspending, inefficiency, and transparency gaps among members of the family.

With technology advancements, computerized expense tracking items were an affordable alternative. Cell phone and web-based software offer automatic, categorized, and graphical accounting of expenses. Existing offerings cover one-person expense tracking and not cooperative management within families. This is a gap since families require systems to track not only spending, but also budget assignment, compliance, savings calculation, and reporting for co-money planning.

Family Expense Tracker bridges this gap by offering a role-based one. The Head of the family can assign budgets monthly, view total expenses, and monitor member expenditure habits. The Members can enter personal spending, view their own expense logs, and monitor spending within the allotted budget. This role-based segregation offers transparency as well as accountability to the household.

Technically, the system is created based on the MERN stack of technologies (MongoDB, Express.js, React.js, and Node.js) because they are modular and scalable and can be used for updation of data in real-time. React.js supports responsive user-friendly UI, and Node.js and Express.js offer a proper backend to guide expense-related functionality. MongoDB elastic NoSQL schema is suitably supported to lead inconsistent financial data such as expenses, categories, and savings. JWT-based authentication also gives protection for access along with suitable role management.

By role-based money management, category-based spending tracking, and comparison of budget versus. savings, and reporting capabilities integrated, Family Expense Tracker is not only easy to use on a day-to-day home finance level but also promotes prudent consumption. In so doing, it promotes enhanced money awareness, decision-making, and overall family long-term financial welfare.

II. LITERATURE REVIEW

Family Expense Tracking Systems (FETS) have evolved from manual bookkeeping and spreadsheet applications to today's dashboard-enabled digital apps with cloud integration and data visualization. Current research has revealed improvements in financial tracking techniques along with challenges like user collaboration, role-based access, and analysis of long-term savings.

Singh and Verma [1] presented a web-based daily expenses and category-focusing personal finance tracker.Their system improved personal budgeting but did not support multiple user roles, limiting its use for family-based applications.

Reddy and Nair [2] created a system for expense management among students that supported data entry and summaries per category. The system, though effective in the generation of visual reports, did not have facilities like budgeting and savings calculation that are important in household finance.

Das and Banerjee [3] have compared the available personal tracking tools such as Mint and YNAB and observed that although these support automation as well as external integrations, they predominantly cover individual use cases and do not provide shared access for families.

MongoDB Inc. [4] points out MongoDB's NoSQL database architecture that supports flexible schema design and hence is appropriate where categories of expenses and users have varied types.

Brown [5] surveyed the move towards cloud and manual finance systems, writing of advantages in terms of instant access and lower error, but citing difficulties of take-up among low-digital-literacy households.

Lee [6] studied the relevance of cloud finance solutions, where the benefits were seen in synchronization and collaboration but raised issues related to data security and internet dependency.

Express.js Foundation [7] describes how Express.js facilitates backend development for financial apps where numerous requests and data exchange need to be processed quickly.

Meta Platforms Inc. [8] elucidates the use of React for creating interactive dashboards using re-usable components and thus rendering it highly appropriate for expense history visualization as well as reporting interfaces.

Node.js Foundation [9] demonstrates how Node.js facilitates scalable, event-driven systems to process multiple expense inputs from various members of a family.

Patel [10] describes the use of JWT authentication in secure financial apps so that sensitive data such as savings and budgets are visible only to the target user.

Sharma [11] talks about incorporating predictive models into spending systems so that trends and predictions can be achieved, which will enhance future versions of family-based financial trackings.

While each of these studies works as an independent framework, they collectively form the basis of the suggested Family Expense Tracker, which utilizes MongoDB, Express.js, React.js, and Node.js to provide a scalable and role-based solution.Compared to existing systems, it incorporates family collaboration, budget planning, savings management, and real-time visualization to counteract the loopholes of existing practices.

III. METHODOLOGY

A. Existing Methodology

Current Family Expense Tracking Systems ("FETS") in both the educational and business communities are generally focused only on the automation of expense recording and allow individuals to have a "one-stop shop" for tracking their expenses. They are primarily offered as either a single-user mobile app or as web-based software, most offer relational databases, and have some basic charts and reports. Even though they are more accurate than traditional hand-entering of expenses, many don't have multi-user collaboration or family-level budgeting while offering limited access based on roles.

[1] Personal Expense Tracker App:

This system replaces old notebooks and spreadsheets with a mobile/web application.

Key Features:

- Web/Mobile Platform: Developed on Android/Flutter for mobile or lower-level web frameworks (browser-based) for client use.
- Cost classification: Cost classification is enabled in the form of food, transport, utilities, etc.
- Reporting: Basic monthly reports and charts can be viewed.
- Authenticated: Email/password with no differentiation of roles.

Limitations:

- Personal use only (do not share with family members).
- Head-level budgeting and monitoring is not even supported with family facility.
- Savings were not being accounted for or monitoring of remaining budget.
- Basic data visualization only - pie/bar chart.
- Limited Reporting/Sharing in WhatsApp and other messaging groups.

[2] Online Expense Trackers (2015-2023):

Online trackers are a useful tool when designed to assist students and business persons with their own financial management.

Key Features:

- Web Portal - Web browser based, a fully desktop ready view and is not meant for mobile usability.
- Categorization / timeline filters - provides the capabilities of expenses to be filtered either by category (i.e., school cumulatively) expense or by the time course (i.e., 2023) expense or time window (i.e., month).
- Reports Download - visual representation of data provided in CSV and/or PDF download.
- Basic Analysing - average monthly spend for certain categories of expense.
- Reports Export - your data into CSV and/or PDF.
- Basic analytics - average spend per month by category.

Limitations:

- Very limited scalability; no hierarchy for family role (i.e., member or head, etc.);.
- Cannot view/compare budget vs. actual.
- Cannot calculate saving or operate accountant intelligence
- Very limited security - no jwt authentication or role gated access
- No expense sharing, or collaborative workflows spanning multiple members.

B. Proposed Methodology

The suggested methodology offers a secure, role-based, modular Family Expense Tracker (FET) application for recording expenses, budget assignment, and reporting to households. The FET identifies 2 distinct roles; Head User members (those who can assign budgets and review recording from all users) and Members (those that create their own expense entries, recording, and viewing for the last assigned budget).

1. System Architecture

The application will be deployed on the MERN stack (MongoDB, Express.js, React.js, Node.js). the overall goal of the stated methodologies will be a scaled secure services solution to financial management

- Frontend: React.js with separate dashboards for Heads & Members
- Backend: Node.js + Express.js for all authentication, authorisation, and business logic.
- Database: MongoDB aims not to impose a schema on expenses, budgets, and roles and provide flexible storage of these service parameters.
- Authentication: login via JWT (JSON Web Token)
- Role-Based Access Control (RBAC): enforced on the backend and the frontend to separate Head & Member authorisation.

2. Functional Modules

- User Authentication & Role Assignment: Heads & Members authenticate via JWT which verifies a member's standing and role allowing heads to set dashboards that are applicable for each member.
- Dashboard Interfaces:
  - Head: Set family budget, track all members' spending, and initiate financial reports with consolidated reporting.
  - Member: Add/update/remove expenses including category, amount, date and description, view a history of expense records and filter by category and/or date
- Expense reporting: Members input expenses with category, amount, date and description; heads are able to see all members' and utilize filtering and summation of totals reported.
- Budgets and Floors: Where members submit their expenses, the system will calculate what is left in the member's budget (savings) after the month is complete, as well as any unclaimed savings by the head.
- Reports and Analytics:
  - Charts: Pie, bar, and line charts to show expenses by category, month, or family member using Chart.js.
  - PDF Reports: Budget and report for the entire family in one download. Includes ₹ symbol support with Noto Sans.

- Sharing Feature: Users can share filtered for budget and spending summary and it can

be shared via WhatsApp to circulate and market the our model.

3. Data Security and Isolation

- Role-Based User Access: Members can only view/manage their own expenses; heads can view/manage all family data.
- JWT Authentication: The system protects all API endpoints and manages user sessions securely and consistently.
- Data Integrity: The application executes rigorous checks to prevent any unauthorized updated to expenses including any attached photos.
- Cloud Deployment: The application uses MongoDB Atlas for secure cloud storage with easy scaling.

IV. SYSTEM DESIGN AND ARCHITECTURE

The Family Expense Tracker (FET) will use a three-tier architecture: Presentation, Application, and Data Layers. This structure will allow the proposed system to be modular, scalable, and maintainable and provide access to each user role (Head and Member).

A. Architectural Diagram

- Presentation Layer (Frontend) - The frontend is developed using React.js, CSS, and Bootstrap creating a responsive and easy-to-use dashboard. The dashboards for both user roles (Head, Member) will be done so that the Head assigns budgets then generates reports, while the Members enter expenses and can view their tracking history.
- Application Layer (Backend) - The backend will be built using Node.js with Express.js to manage the business logic of the application regarding API routing, Authentication, Authorization, calculations for budget, Savings, and report generation.
- Data Layer (Database) - MongoDB will be used to store the data of Expenses, Users' profiles, Budgets, and Report records. MongoDB is a NoSQL document store database, this means the unstructured collection will allow querying quickly and has the ability to scale the records, even with multiple family members.
- Communication Flow: Layers communicate using data and requests are encrypted and if getting data from the API, all API request will require an authorization check for each transaction with JWT assigned for session validation.

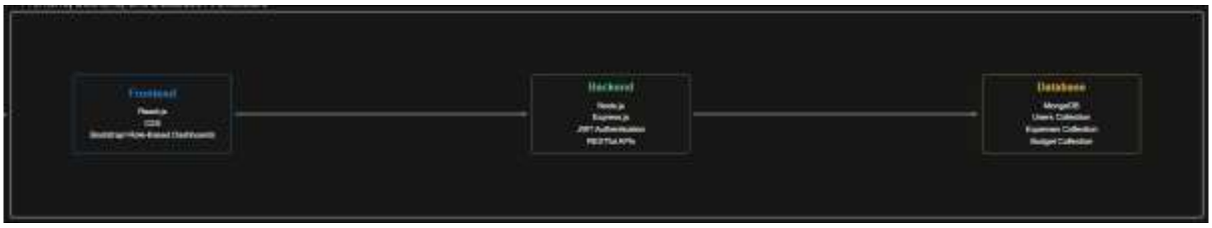


Fig. 1: System Architecture Diagram

B. Security and Scalability Considerations

- Security: The system has implemented Role-Based Access Control (RBAC) in both the frontend and backend where a Head can see the family aggregate data or edit it while a Member can only see their expense data. In the future, we will log activities around any of the more sensitive transactions (for example, the budget allocations and Expense Reports). User sessions will have explicitly short-lived JWT for security.
- Scalability: As a result of the modular aspect of MERN, the frontend, backend and the database can each be scaled. Additionally, deployment productivity shall be improved using Docker containerizations or cloud vendor auto-scaling for AWS, Azure or GCP, etc.
- Extensibility: Without changing the architecture of the Family Expense Tracker, the addition of other features for example, multi-currency support or expense forecasting (ML), or a Payment Gateway can be added into Family Expense Tracker.

C. Data Flow Diagram (DFD):

The Data Flow Diagram (DFD) diagram depicts how data flows through the Family Expense Tracker, and depicts how the user roles (Head and Member) interact with the backend and the database.

Key Components and their Relation to Each Together

- Head:
    - Logs into application.
    - Allocates budgets to families.

  - Lists family aggregate reports of expenses.
  - Tracks, via each member, category of expense, period/month.
- Member:
  - Logs into application.
  - Adds an expense (expense category, amount, description, date).
  - Searches/edits/deletes own expense history.
  - Filters results and shares to WhatsApp.
- Database Layer
  - Stores roles, budgets, expense history and savings.
  - Filters on role (i.e. Heads see all data, Members only see personal).

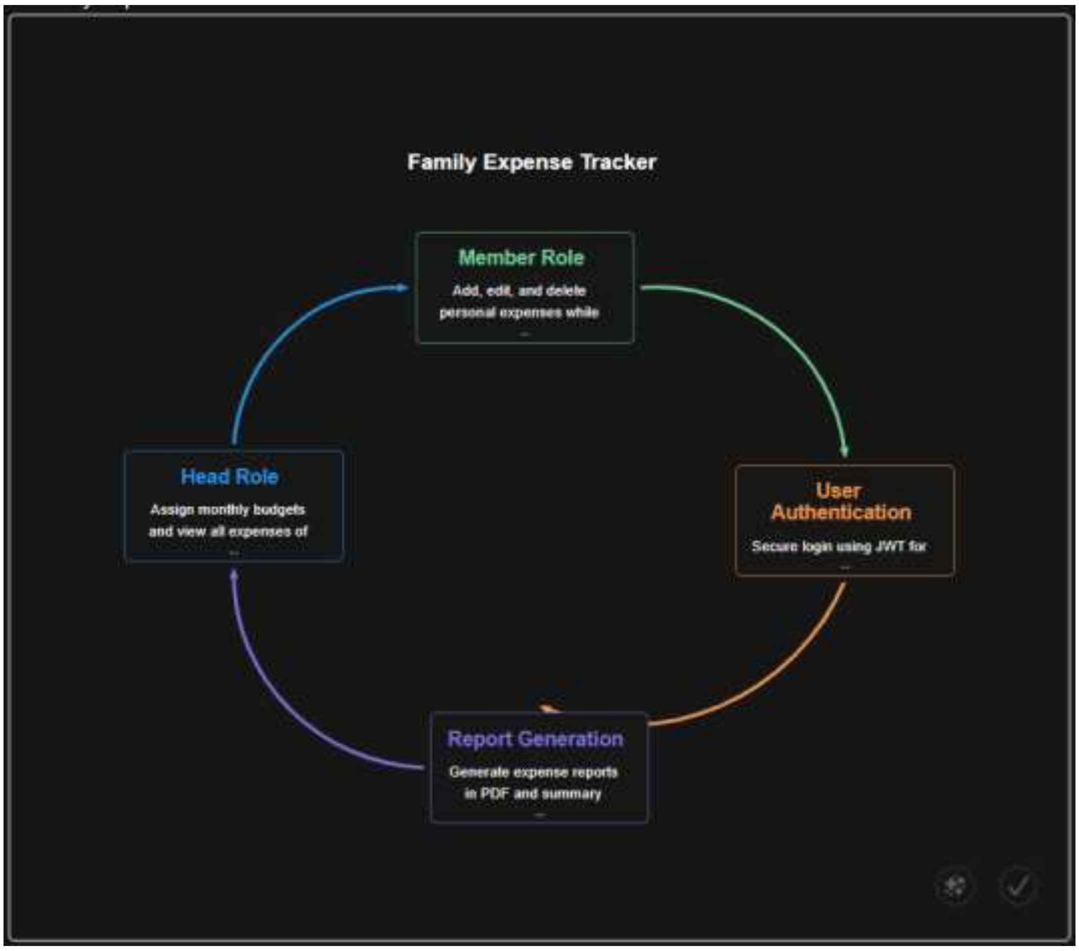

Family Expense Tracker
Member Role
Add, edit, and delete personal expenses while ...
User Authentication
Secure login using JWT for ...
Report Generation
Generate expense reports in PDF and summary ...
Head Role
Assign monthly budgets and view all expenses of ...

Fig. 2: Data Flow Diagram

V.IMPLEMENTATION

A. Frontend Implementation

The client is also written in React. js, CSS and BootStrap is used for responsive design and user friendly.

Key frontend features include:

- Role-Based Dashboards: Individual dashboards for Head and Members with certain role based features.
- jQuery Converted To React Components : All chat box, table, modals chart are converted to react components as reused.
- RESTful API Integration: Consumed secure data from backend RESTful APIs and Axio.
- Form Validations: Uses Html5 validation with your own JavaScript validation so the user can\'t submit the form with bad input.

- Charts & Analytics: You can Chart the Expense data. js + react-chartjs-2 (Pie, Line, Bar).

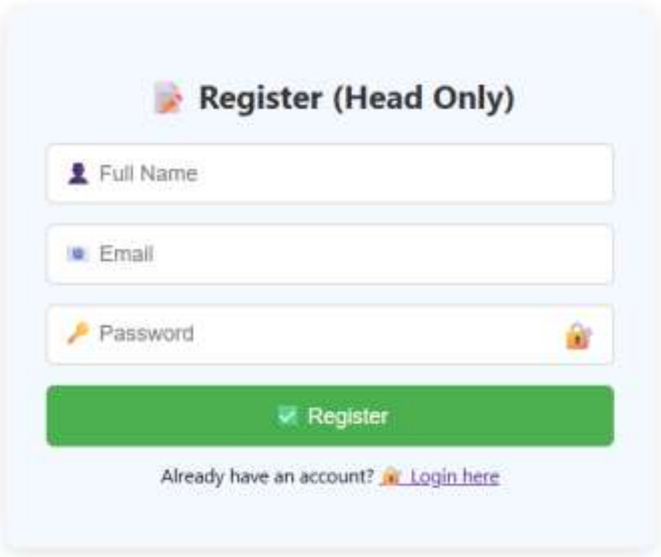


Fig. 3: Register Page for Head of the Family

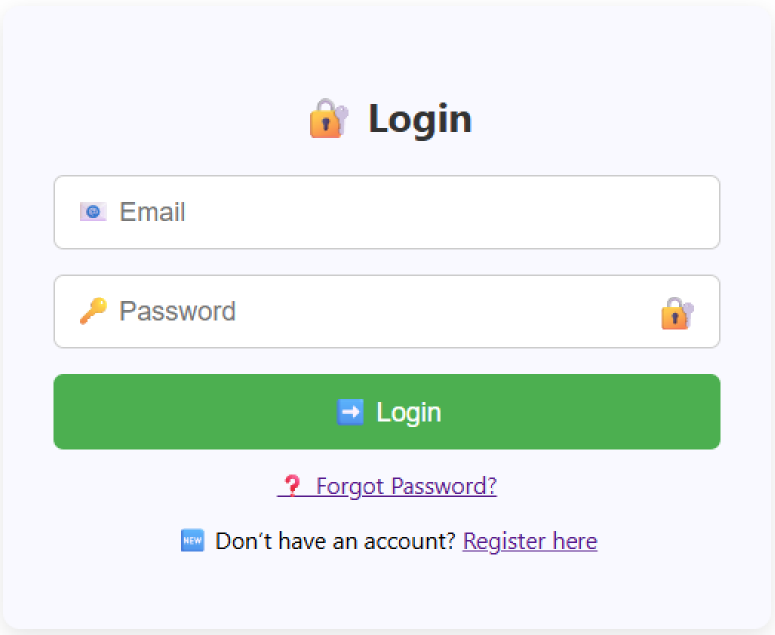


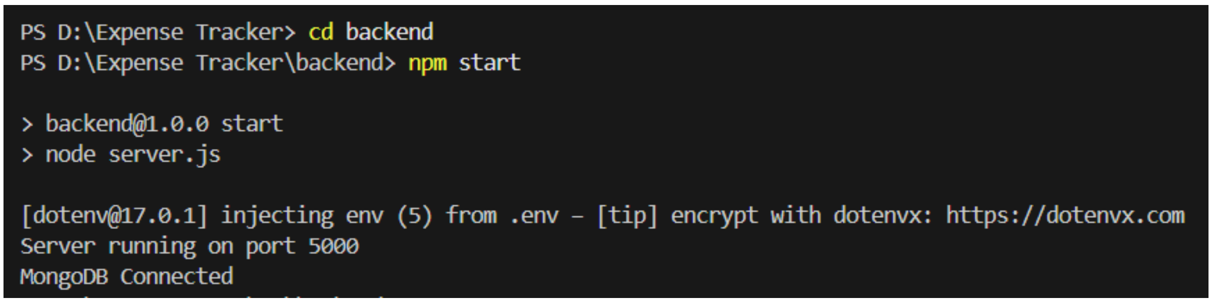
Fig. 4: Login Page for Member Dashboard and Head Dashboard

B. Backend Implementation

On the backend, that's Node. js and Express. js and it is MVC (Model-View-Controller) framework.

Backend features include:

- Models: Define user schemas and expense, and also budget.
- Controllers – Command requests, business rules, response types.
- Routes: Define your API URIs by business logic API routes, guard with auth and role-based middleware.
- Authentication: All protected routes in this application (including the ACTION Creators and ACTION Reducers) are secured with JWT authentication.
- Error Handling: It also comes with your average error-handling middleware so you don't have to manage error responses on your own.



```
PS D:\Expense Tracker> cd backend
PS D:\Expense Tracker\backend> npm start

> backend@1.0.0 start
> node server.js

[dotenv@17.0.1] injecting env (5) from .env – [tip] encrypt with dotenvx: https://dotenvx.com
Server running on port 5000
MongoDB Connected
```

Fig.5: Backend Server Running Successfully

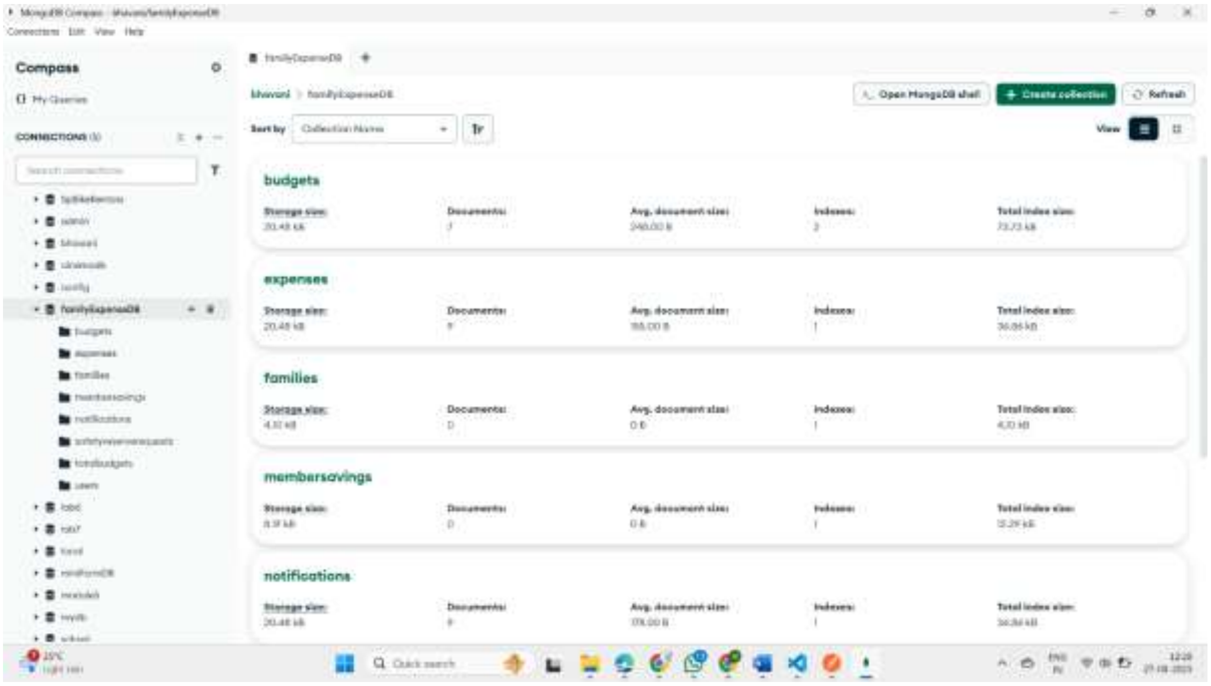
C. Database Implementation

We realize the database in MongoDB, and it adopts a flexible document schema to store both structured data and unstructured data.

Collections include:

- Users Collection: This will hold user details, roles(Head/Member) and Family Id’s.
- Expenses Collection: category, amount, date, and description for each expense in the Stores.
- Budgeting Grouping: Saves family budget, member by member budget, and received savings.

An index is created on the userId and the date to facilitate read-operation.



budgets
Storage size: 20.48 kB Documents: 7 Avg. document size: 248.00 B Indexes: 2 Total index size: 73.73 kB

expenses
Storage size: 20.48 kB Documents: 9 Avg. document size: 188.00 B Indexes: 1 Total index size: 36.86 kB

families
Storage size: 4.10 kB Documents: 0 Avg. document size: 0 B Indexes: 1 Total index size: 4.10 kB

membersavings
Storage size: 8.19 kB Documents: 0 Avg. document size: 0 B Indexes: 1 Total index size: 12.29 kB

notifications
Storage size: 20.48 kB Documents: 9 Avg. document size: 178.00 B Indexes: 1 Total index size: 36.86 kB

Fig. 6: MongoDB Compass View of expense_tracker collections

D. Key Modules Implemented

1. User Management Module
   - Head can register/login.
   - Users can log in securely via JWT.
2. Expense Management Module
   - Users may add, edit, remove expenses.
   - Category, date and month filters.
   - Filter by amount, type, or date.

- Share filtered results via WhatsApp.

3. Budget Management Module
    - Head assigns a monthly family budget and member budgets.
    - Automatic savings generated from remaining balances.
4. Reporting & Analytics Module
    - Head can generate member-wise, category-wise, monthly reports.
    - Design PDF Reports with ₹ symbol (font Noto Sans).
    - Interactive Pie Chart, Line Chart and Bar Chart.

E. Testing and Validation

- Unit Testing: Same would be good -Your good to write the test also – Common Approach is primarily followed to do the unit testing to test the API (backend) using Postman-Requset and Jest.
- Functional Testing: Tested the flows of expense entry, budget imbibition and report creation.
- Role-Based Testing: Confirmed that Head and Members received the appropriate authorized data.
- Multi-Devices Testing: Confirmed to work perfect on desktop, tablet and any other smartphones.

VI. TECHNOLOGY AND STACK OVERVIEW

Family Expense Tracker (FET) is built on the MERN stack, i.e., MongoDB, Express.js, React.js, and Node.js.

The full-stack JavaScript environment allows for simple communication between frontend, backend, and database and also is lightweight and scalable.

A. MongoDB

MongoDB is a NoSQL database where data are stored in loose, JSON-like documents instead of tight relational tables.

Benefits for FET:

- Schema flexibility to support heterogeneous user expenses, budgets, and savings.
- High replication and scalability for storing the entire family's data.
- Quick data crunching and instant queries—that's how you whip up reports in no time.

How it works in FET: Basically, every bit of info—user profiles, spending, budgets, savings—gets tossed into collections like Users, Expenses, Budgets, and Savings. All neat and tidy.

B. Express.js

Express.js, if you haven't met, is this featherweight champ of a Node.js web framework, perfect for cranking out RESTful APIs.

Why bother with it in FET?

- Handles stuff like user authentication, permissions, and oh—the inevitable errors.
- Routes? Covered. It manages expense, budget, and user APIs, scrubbing input so nothing sketchy gets through.
- Can juggle a whole crowd of user requests without breaking a sweat.

How FET uses it: It's the gatekeeper for request checks, JWT logins, and basically turns whatever the frontend throws at it into stuff the database actually understands.

C.React.js

React.js is a JavaScript library for rendering dynamic browser UI.

FET benefits:

- Reuse of form, table, and dashboard via component-based modeling.
- faster rendering of charts and reports using Virtual DOM.
- Supports Bootsrap for responsive design.

Applications in FET: Builds role-based dashboards (Head/Member), expense forms, budget assignment screens, and analytics charts.

D. Node.js

Node.js is a non-blocking, event-driven runtime for running JavaScript beyond the browser.

FET benefits:

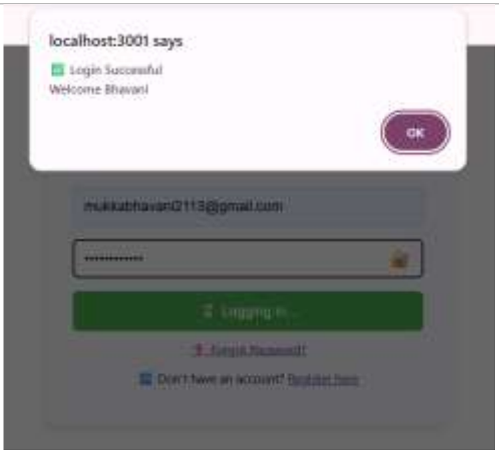
- Takes multiple user expense submissions effectively.
- frauen Best suited for real-time budget updates and notifications.
- Large NPM ecosystem to include libraries like Chart.js.

Usage in FET: Runs backend logic, handles API requests, and MongoDB integration with async tasks.

E. Other Tools and Libraries

- Bootstrap + CSS: For mobile-responsive uniform styling.
- Axios: For frontend-backend communication via REST APIs.
- Chart.js + react-chartjs-2: Employed to generate Pie Charts, Line Graphs, and Bar Graphs for financial visualization.
- JSON Web Tokens (JWT): Employed for secure authentication and session management.
- MongoDB Atlas: Employed for cloud storage, backup, and high availability.

VII. RESULTS



localhost:3001 says

✅ Login Successful
Welcome Bhavani

OK
mukkabhavani2113@gmail.com
Logging in...
Forgot Password?
Don't have an account? Register here

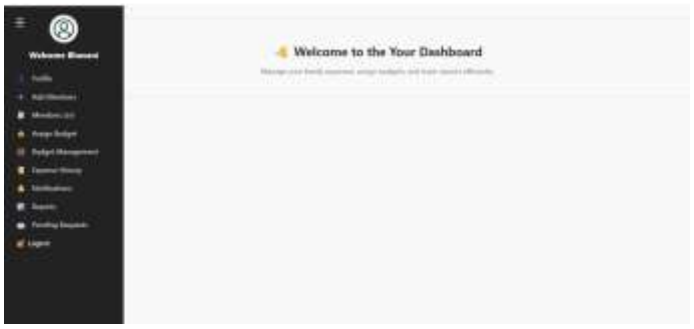
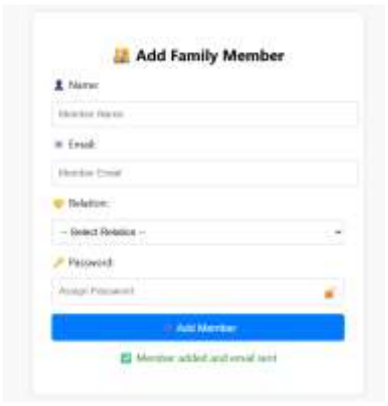
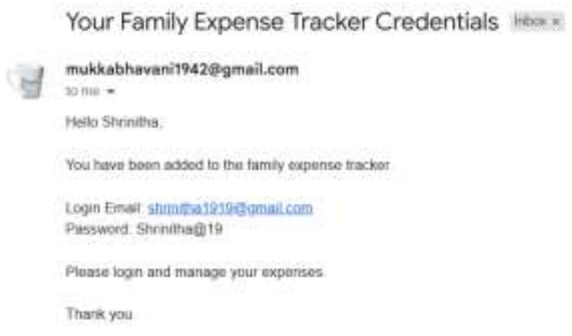


Fig. 7:Head login successfully

Fig. 8: Head dashboard



✅ Member added and email sent



Your Family Expense Tracker Credentials Inbox ×

mukkabhavani1942@gmail.com
to me

Hello Shrinitha,

You have been added to the family expense tracker

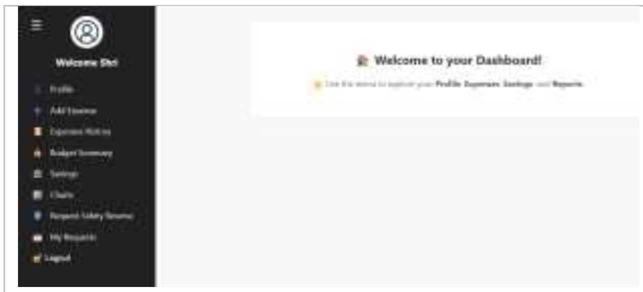
Login Email: shrinitha1919@gmail.com
Password: Shrinitha@19

Please login and manage your expenses

Thank you

Fig. 9:Head adds family member

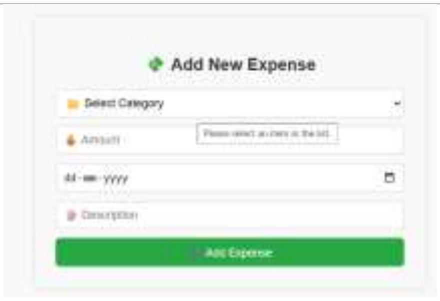
Fig. 10: Member receives credentials through mail


Welcome Shri

Profile
Add Expense
Expenses History
Budget Summary
Settings
Charts
Request Safety Reserve
My Requests
Logout

Welcome to your Dashboard!

Use the menu to explore your **Profile**, **Expenses**, **Settings** and **Reports**.


Add New Expense

Select Category

Amount — Please select an item in the list.

dd-mm-yyyy

Description

Add Expense

Fig. 11:Member Dashboard Fig. 12:Member can add expenses

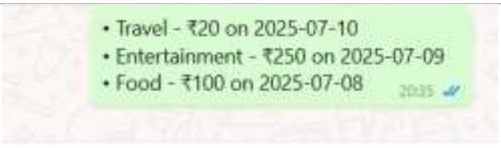
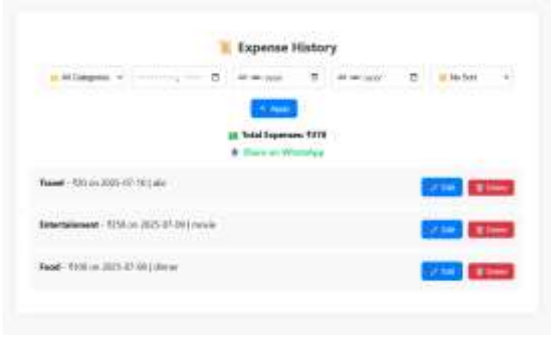

• Travel - ₹20 on 2025-07-10
• Entertainment - ₹250 on 2025-07-09
• Food - ₹100 on 2025-07-08 20:35

Fig. 10:Export Through Pdf


Expense History

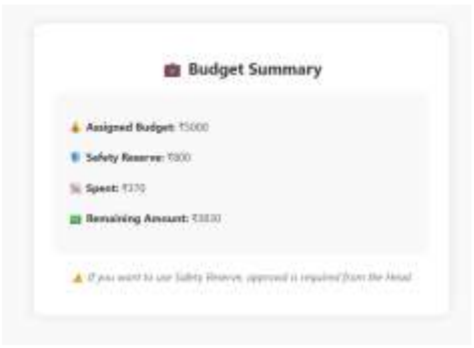
All Categories | dd-mm-yyyy | dd-mm-yyyy | dd-mm-yyyy | No Sort

Apply

Total Expenses: ₹370

Share on WhatsApp

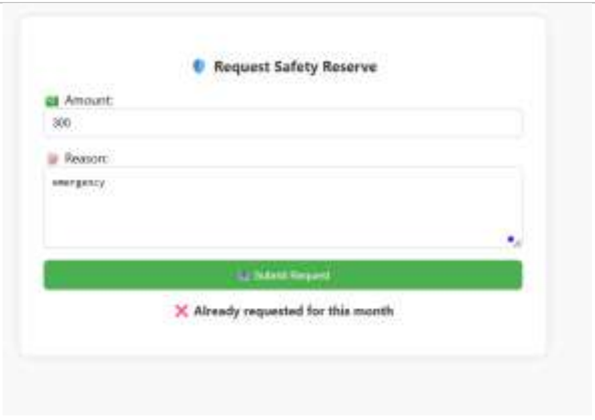
Travel - ₹20 on 2025-07-10 | abc — Edit | Delete
Entertainment - ₹250 on 2025-07-09 | movie — Edit | Delete
Food - ₹100 on 2025-07-08 | dinner — Edit | Delete


Budget Summary

Assigned Budget: ₹5000
Safety Reserve: ₹800
Spent: ₹370
Remaining Amount: ₹3830

If you want to use Safety Reserve, approval is required from the Head.

Fig. 13: Member Expense History Fig. 14:Member Budget Summary


Request Safety Reserve

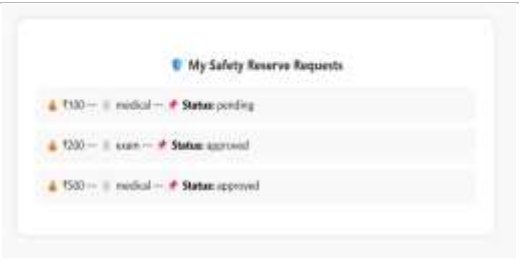
Amount:
300

Reason:
emergency

Submit Request

Already requested for this month

Fig. 15: Member requests an amount from the safety reserve to the head


My Safety Reserve Requests

₹130 — medical — **Status:** pending
₹200 — exam — **Status:** approved
₹500 — medical — **Status:** approved

Fig. 16: Member safety reserve requests

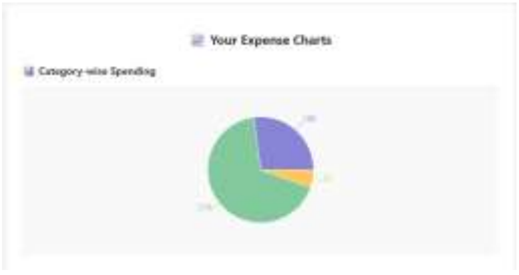
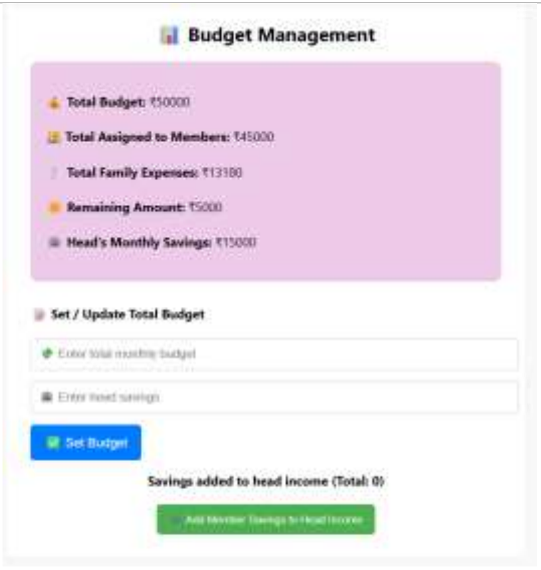


Fig. 17: Member Expense History in visual represenatation



Budget Management

Total Budget: ₹50000

Total Assigned to Members: ₹45000

Total Family Expenses: ₹13180

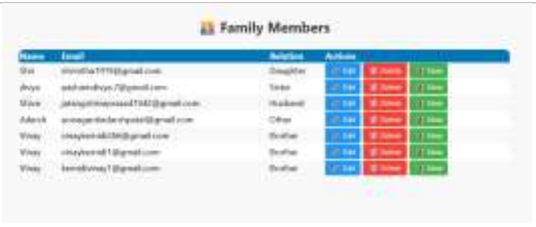
Remaining Amount: ₹5000

Head's Monthly Savings: ₹15000

Set / Update Total Budget

Savings added to head income (Total: 0)

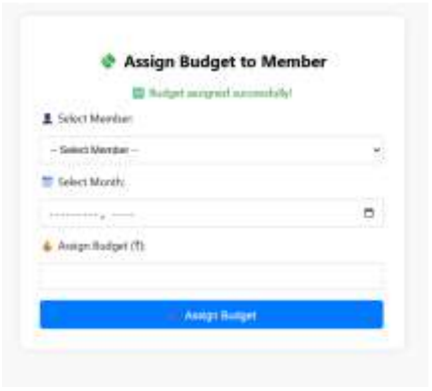
Fig. 18: family budget assigned by head



Family Members

Name	Email	Relation	Actions		
Shri	[illegible]	Daughter	Edit	Delete	View
divya	[illegible]	Sister	Edit	Delete	View
Shiva	[illegible]	Husband	Edit	Delete	View
Adarsh	[illegible]	Other	Edit	Delete	View
Vinay	[illegible]	Brother	Edit	Delete	View
Vinay	[illegible]	Brother	Edit	Delete	View
Vinay	[illegible]	Brother	Edit	Delete	View

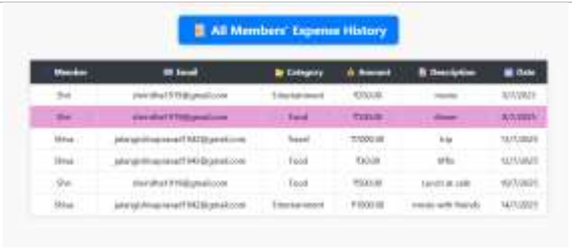
Fig. 19: Family Members List



Assign Budget to Member

Budget assigned successfully!

Fig. 20: Head assigning budget to family member



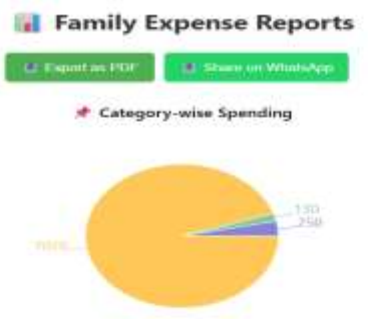
All Members' Expense History

Member	Email	Category	Amount	Description	Date
Shri	[illegible]	Entertainment	[illegible]	movie	[illegible]
Shri	[illegible]	Food	[illegible]	dinner	[illegible]
Shiva	[illegible]	Travel	[illegible]	trip	[illegible]
Shiva	[illegible]	Food	[illegible]	tiffin	[illegible]
Shri	[illegible]	Food	[illegible]	lunch at cafe	[illegible]
Shiva	[illegible]	Entertainment	[illegible]	movie with friends	[illegible]

Fig. 21: All members Expense History

Alerts & Notifications

Fig. 22: Alerts for inactivity and high spendings



Family Expense Reports

Fig. 23: Category wise spending

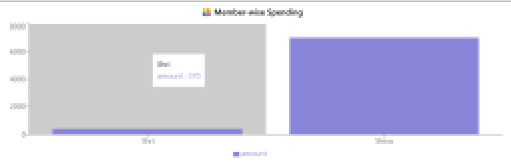
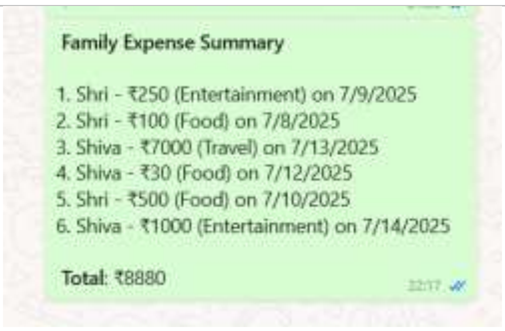


Fig. 8: Member-wise spending

Monthly Expense Trend

Fig. 24: Monthly-wise spending

Fig. 25: Export as Pdf



Family Expense Summary

1. Shri - ₹250 (Entertainment) on 7/9/2025
2. Shri - ₹100 (Food) on 7/8/2025
3. Shiva - ₹7000 (Travel) on 7/13/2025
4. Shiva - ₹30 (Food) on 7/12/2025
5. Shri - ₹500 (Food) on 7/10/2025
6. Shiva - ₹1000 (Entertainment) on 7/14/2025

Total: ₹8880

22:17

Fig. 26: Export through whatsapp

VIII. DISCUSSION

The Family Expense Tracker was designed to make money management for families and their everyday expenditure and savings easier. By means of testing, it performed excellently to enable the household head to prepare budgets for all, track their expenditure, and even calculate their savings at the end of each month without even them lifting a finger. This added organization and transparency in family finance management.

Both the head and member dashboards were simple to navigate through, and the pie, bar, and line charts helped to easily see where money was spent. When compared to writing on notebooks and spreadsheets, the system saved time and fewer errors occurred because real-time feedback was given.

The project also demonstrated that the utilization of the MERN stack provides flexibility and scalability, thus the system will be able to accommodate more users or data later on. Security was implemented through role-based access as well as JWT authentication, securing financial data.

Nonetheless, there are few constraints. Currently, there is no app for mobile and no provision of smart suggestions or linking with online payment schemes. Incorporating the latter two in the near future will make the system even more valuable to use.

In conclusion, Family Expense Tracker is a simple yet effective way of managing family savings, expenses, and budgets and making planning easier for families.

VIII. CONCLUSION

The Family Expense Tracker (FET) illustrates the application of contemporary full-stack web technologies to make and automate family-level financial management easy. By developing the system using the MERN stack (MongoDB, Express.js, React.js, Node.js), the solution provides a role-based, scalable, and secure platform that enables Heads and Members to be empowered.

Compared to other existing personal expenditure trackers, FET has role-based budgeting, auto-save logic, and visual analytics at an overview level, which more suit households.

JWT authentication and Role-Based Access Control (RBAC) guarantee safe handling of financial data.

The system not only simplifies budgeting and tracking of expenses but also facilitates better decision-making with organized reports and visual insights.Usability testing confirmed its ease, responsiveness, and usability.

FET in the future can be made more feature-rich with multi-currency support, AI prediction, and mobile apps, making it even more applicable to real life. This project is a tangible demonstration of full-stack development on fiscal matters at the home and family scale, offering fiscal literacy and utilization of resources.

REFERENCES

[1] Ganesh, C., & Poornima, S., “MERN Stack Finance Tracker with Real-Time Analytics,” *International Journal of Scientific Research and Engineering Development*, vol. 8, no. 3, 2025. Available: https://ijsred.com/volume8/issue3/IJSRED-V8I3P423.pdf

[2] Shubhangi Bhardwaj, Sneha Gupta, Unnati Jaiswal, & Riya Bhargava, “Personal Expense Tracker,” *International Journal of Novel Research and Development*, vol. 9, no. 5, 2024. Available: https://www.ijnrd.org/papers/IJNRD2405499.pdf

[3] Bekaroo, G., & Sunhaloo, M. S., “Intelligent Online Budget Tracker,” *ResearchGate*, 2007. Available: https://www.researchgate.net/publication/237448489_Intelligent_Online_Budget_Tracker

[4] Maravi, A., & Dewangan, O., “Expense Tracker Application (Android-Based),” *International Journal of Novel Research and Development*, vol. 9, no. 5, 2024. Available: https://www.ijnrd.org/papers/IJNRD2405409.pdf

[5] IJSRED, “Expense Tracker with In-built AI,” *International Journal of Scientific Research and Engineering Development*, vol. 8, no. 2, 2025. Available: https://www.ijsred.com/volume8/issue2/IJSRED-V8I2P397.pdf