Expense Tracker Application

Digital Transformation 2

Krishna Gupta Sarim Rouf Khan Yussuf Hassan Hamad

IIT Madras Zanzibar

December 21, 2025

Introduction & Problem Statement

- Manual expense tracking is inefficient and error-prone
- Users lack visibility into spending patterns
- No forecasting of future expenses

Proposed Solution

- Digital expense logging
- Visual analytics
- AI-based spending prediction

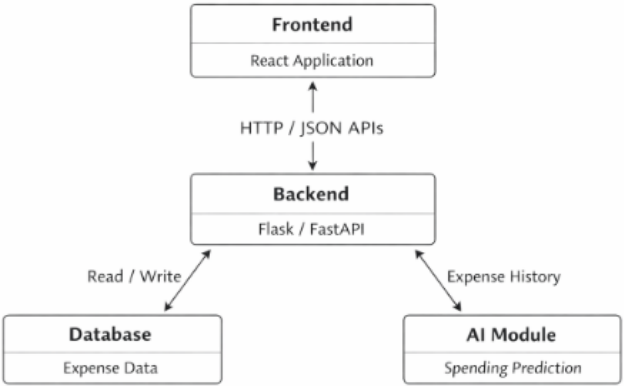
Target Users & Need

- Students managing monthly budgets
- Working professionals
- Budget-conscious individuals

Why This Solution is Needed

- Improves financial awareness
- Helps control overspending
- Enables data-driven decisions

System Architecture

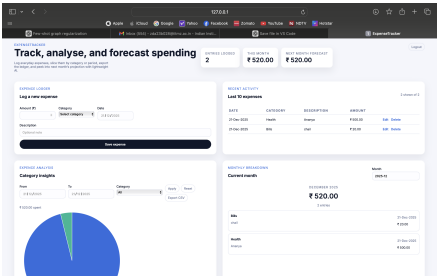


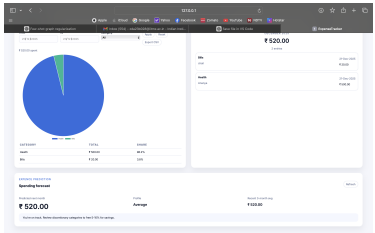
Frontend communicates with backend APIs. Backend manages data storage and AI prediction.

Frontend Architecture

Technology: React.js

- Add Expense Form
- Recent Expenses List
- Edit / Delete Expense
- Charts Dashboard
- Responsive UI





Backend Architecture

Technology: Flask / FastAPI

- RESTful API design
- Input validation
- Business logic
- AI model integration

Backend API Code Snippet

Add Expense Endpoint

```
@app.post("/expenses")
def add_expense(expense: Expense):
    db.insert(expense)
    return {"message": "Expense added successfully"}
```

Database Design

Expense Table Schema

- id (Primary Key)
- amount
- category
- date
- description
- user$_id$

One user can have multiple expense records.

AI Integration

Model Used: Linear Regression

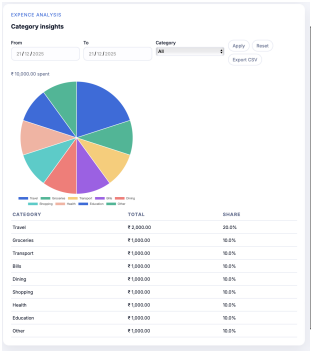
- Uses last 3 months of expense data
- Predicts next month spending
- Categorizes user spending behavior

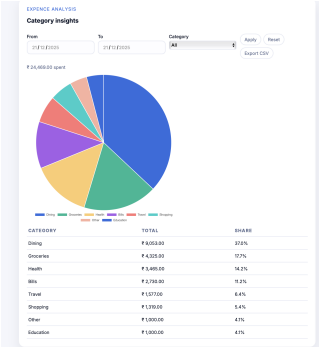
AI Model Code Snippet

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
prediction = model.predict(X_test)
```

Application Outputs





The following charts and plots provide a visual representation of the data, highlighting key trends, patterns, and comparisons to support the analysis and insights.

Live Demo Flow

1. Add expense
2. View recent expenses
3. Edit / delete expense
4. View charts and statistics
5. Check next month prediction

Backup demo video available in case of technical issues.

Challenges & Learnings

Challenges

- Accurate prediction with limited data
- Validation at all API endpoints

Learnings

- Full-stack application development
- REST API design
- AI integration in real-world systems
- Docker-based deployment

Conclusion & Future Scope

Conclusion

- Complete expense tracking solution
- Analytics and AI prediction

Future Enhancements

- User authentication and authorization
- Bank API integration
- Budget alerts and notifications
- Advanced machine learning models

Thank You!!