Expense Tracker
API Documentation

Krishna Gupta Sarim Rouf Khan Yussuf Hassan Hamad

December 21, 2025

1. Introduction

The Expense Tracker application provides a set of RESTful APIs that allow users to record, retrieve, and manage their expense data efficiently. The backend processes user requests, performs validation, and stores data persistently.

2. Base URL

`http://localhost:5000/`

3. API Endpoints

3.1 Add Expense

Endpoint:

`POST /add-expense`

Description: Adds a new expense entry to the system.
Request Body (JSON):

```
{
  "date": "2025-03-20",
  "category": "Food",
  "amount": 250,
  "description": "Lunch"
}
```

Validation Rules:

- Date must be valid
- Category cannot be empty
- Amount must be a positive number

Success Response:

```
{
  "message": "Expense added successfully"
}
```

3.2 Get All Expenses

Endpoint:

`GET /expenses`

Description: Fetches all stored expense records.
Success Response:

```
[
  {
    "id": 1,
    "date": "2025-03-20",
    "category": "Food",
    "amount": 250,
    "description": "Lunch"
  }
]
```

3.3 Delete Expense

Endpoint:

`DELETE /delete-expense/<id>`

Description: Deletes an expense record using its unique identifier.
Path Parameter:

- `id`: Expense ID

Success Response:

```
{
  "message": "Expense deleted successfully"
}
```

3.4 Expense Summary (Optional)

Endpoint:

`GET /summary`

Description: Returns total expenditure and category-wise breakdown.
Success Response:

```
{
  "total": 350,
  "categories": {
    "Food": 250,
    "Transport": 100
  }
}
```

4. HTTP Status Codes

Code	Description
200	Success
400	Bad Request
404	Not Found
500	Internal Server Error

5. API Workflow

1. Frontend sends HTTP request
2. Backend validates input
3. Business logic executes
4. Data is stored or retrieved
5. JSON response returned

6. Conclusion

The Expense Tracker API follows REST principles to provide reliable and scalable expense management functionality. It enables seamless communication between frontend and backend components using JSON-based data exchange.