**Requirement Gathering and Analysis Phase**

**Technology Stack (Architecture & Stack)**

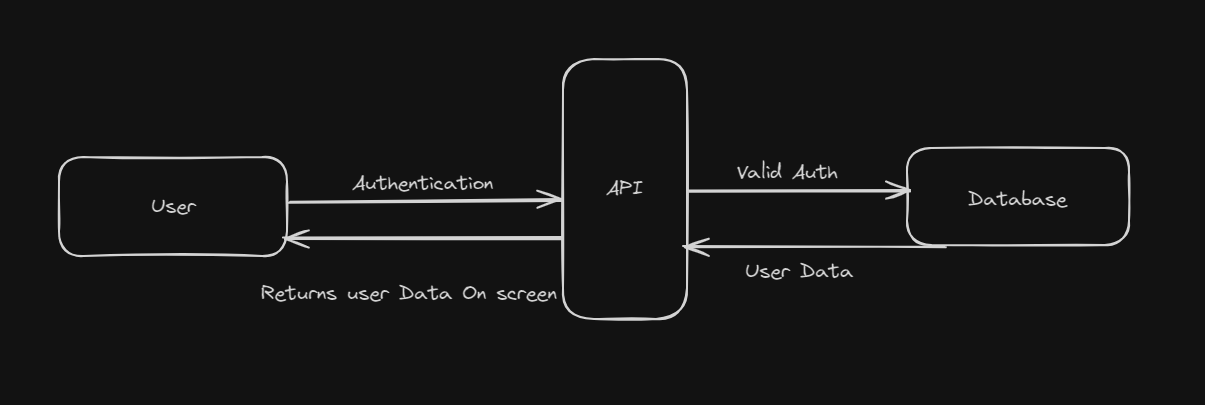
|  |  |
| --- | --- |
| Date | 29 June 2024 |
| Team ID | SWTID1720004076 |
| Project Name | Wise Wallet: Your Budgeting Partner |
| Maximum Marks |  |

# **Technical Architecture:**

## FinTrack: EXPENSE TRACKER APP

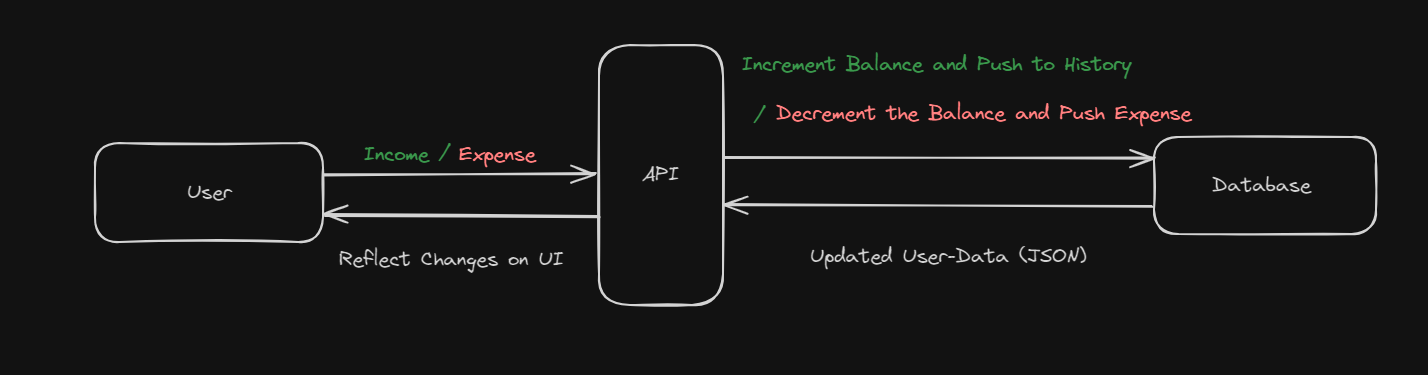
The following architectural diagrams discuss different use cases of the Expense tracker web-app.

1. User Authentication



* An invalid authentication will lead to a Bad Authentication Request (Status: 403)

1. Adding Transactions



* These expenditures could be shown both monthly as well as daily.

**Table-1: Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No | Component | Description | Technology |
|  | UI (User Interface) | Client-Side Interaction (Web-App Frontend) | ReactJS (v18) |
|  | API (Not external) | Server-Side to obtain data from DB | NodeJS + ExpressJS |
|  | NoSQL Database | Database to store information of user such as, Current balance, expenses, Income, etc. | MongoDB Atlas |
|  | Backend Testing | To test all routes and controller functions | Postman |
|  | Version Control | To keep a log of all versions of the web-app | GitHub, Git |
|  | Deployment | In order to deploy the web-app after it is production ready | Vercel/Netify |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No | **Characteristics** | **Description** | **Technology** |
|  | **Authentication Security** | **Security while users log-in/sign-up, can login with google** | **JWT Auth and Cookies.** |
|  | **User-Interface Performance** | **To ensure an extremely responsive and high performing page that has high scores in FCP, LCP and TBT** | **SSR in ReactJS (v18)** |
|  | **Scalability** | **The following is a simple web-app that is not very server heavy** | **Regular 3rd tier** |