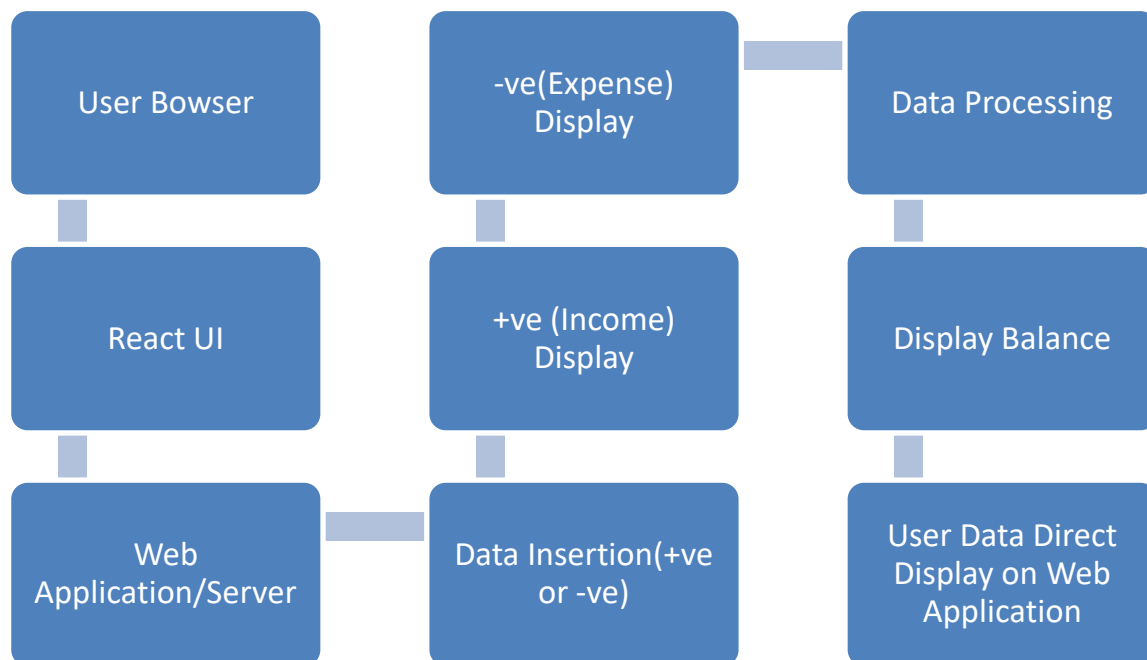# Money Tracker Architecture

Architecture Consist on following Steps:-

- User browser
- React UI
- Web application or server
- Data insertion
- +ve Income Display
- -ve Expense Display
- Data Processing
- Display Balance
- User Data Display on Front-end

User Bowser	-ve(Expense) Display	Data Processing
React UI	+ve (Income) Display	Display Balance
Web Application/Server	Data Insertion(+ve or -ve)	User Data Direct Display on Web Application

Description:-

Each step in the architecture outlined:

1. User Browser:

   - This is where the user interacts with your web application. Users input data, interact with the user interface, and make requests to your server.

2. React UI:

   - React is a JavaScript library for building user interfaces. In this step, you are using React to create the front-end of your web application. It handles the presentation and user interaction components, making the user interface dynamic and responsive.

3. Web Application or Server:

   - This is the backend of your application. It can be a web server or an application server responsible for handling requests from the React UI, processing data, and returning responses. It typically consists of server-side code that can include routing, authentication, and business logic.

4. Data Insertion:

   - This step involves taking user inputs or data submitted via the React UI and inserting it into a database or storage system. This data could include income and expense records, which are essential for tracking finances.

5. +ve Income Display:

   - After inserting income data into your storage system, you'll need to retrieve and display it in a user-friendly format on the React UI. This display could show details such as income source, amount, date, and any other relevant information.

6. -ve Expense Display:

- Similar to displaying income data, you'll also need to retrieve and display expense data from your storage system. This display should include details like the expense category, amount, date, and any associated information.

7. Data Processing:

   - In this step, your server may perform calculations or data processing tasks. For a financial application, this could involve calculating the user's overall balance, which is the result of income minus expenses.

8. Display Balance:

   - After processing the data, you should display the user's balance on the React UI. This gives the user a clear picture of their financial situation, showing them how much money they have available.

9. User Data Display on Front-end:

   - This step involves displaying additional user-specific data on the front-end, such as profile information, account settings, or transaction history. This data may be retrieved from the server and presented to the user.