**ADP Innovation Labs Assignment**

# Application Overview

Below is the high-level design diagram of the application. Application has three tiers. UI Tier is responsible for displaying a list of top earner transactions for the selected year. Middle Tier is responsible for request validation, handles server errors from source and error logging. Source Tier providers employee data and store transactions. HTML page for the UI tier is hosted using vanilla NodeJS. Middle Tier runs using NodeJS and http.

Diagram

Description automatically generated

## UI Tier

This tier is responsible for viewing employee data and handling user actions.

Technology stack:

1. HTML
2. CSS
3. JavaScript
4. Web API Fetch

### Choose Year

This is a dropdown with predefined years. For the assignment purpose I have kept only year 2021.

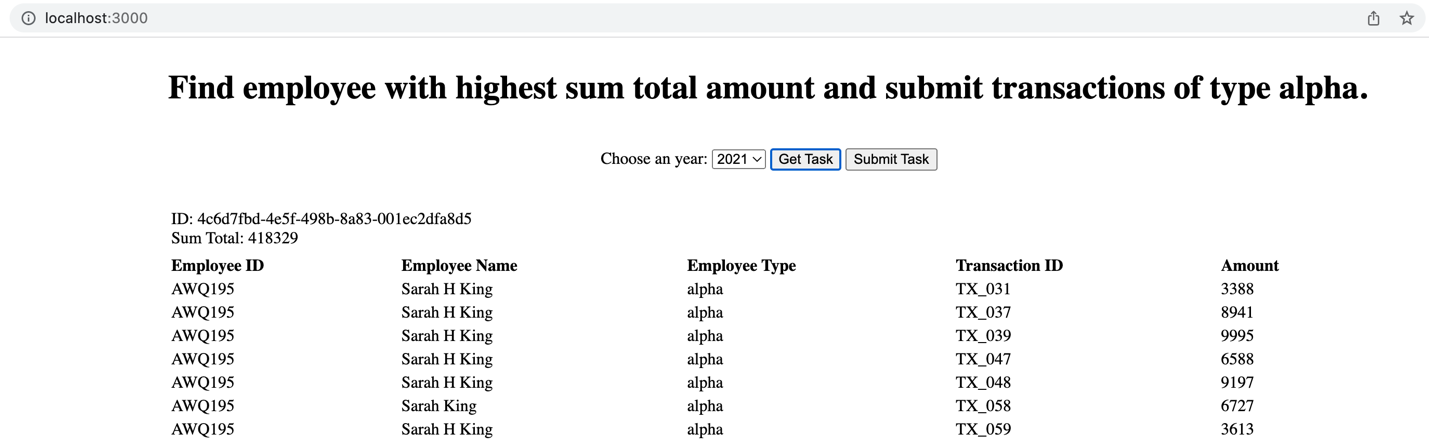
### Get Top Earner

Function that sends GET request for finding top earner. Once the result is received it will list employee details in a table. Also, display the ID and sum on top of the table.

### Submit Task

Function will frame the request body and sends POST request to submit the task. This function will display success or error message depending on the status code.

There are few other utility functions to handle repetitive tasks. Below are few screen shots from the application which is capable to handle responsive view and user-friendly error messages.



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

## Middle Tier

This tier has two main responsibilities. It serves the HTML file to browser and handles API request from UI. This layer is also responsible for validating request parameters, handling errors from source tier and appropriate error logging. There are three modules in this tier. App.js will initialize the server and listens to the default port 3000. It takes care of API routes and the validation of input parameters. It also renders the HTML page for the default route. HTML page is added in the views folder. Controller is responsible for handling API actions.

Libraries Used:

1. NodeJS
2. Http
3. fs
4. Axios

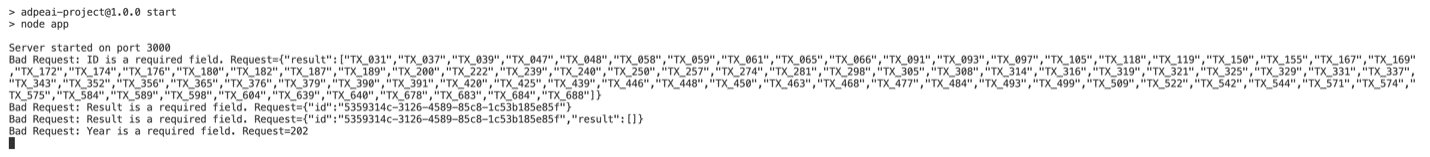
### Top Earner

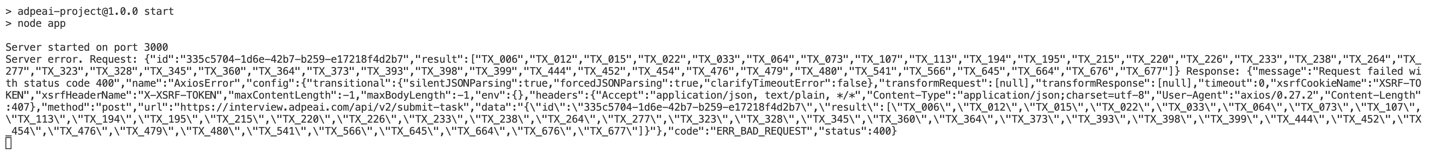
This is a GET API which takes the year as a URL parameter. It will then get the task from source tier and filter the data using the input year parameter. Once the data is received it will find the total sum for each employee. Total sum for each employee is stored as a key value pair where key as the employee id. Then it will find the max sum and employee id corresponds to max sum. Finally, the transactions are filters with the employee id having highest sum and type alpha. These results are returned in API response.

### Submit Task

This is a POST API and expects id and result in request body. It will make sure the incoming request has an id and result array with transaction ids. For a valid request this function will call the backend API to post the request and return an object with message and status code.

Middle Tier has a good error logging mechanism. It will console log request and response along with error message. Following are few screenshots of errors logged on the terminal.





## Commands to run the application

npm install

npm start

URL: <http://localhost:3000/>

GitHub Repository: <https://github.com/manojr0126/adpeai-project>