

GitHub URL

- https://github.com/sr1983-it-gl/07_GL_FE_May_2023_Participants/tree/main/08-Lab-04/Expense-Tracker
-

Tasks

- Creation of the project folders
 - expense-tracker-app
 - client
 - Server
- Setup the client project
 - “npx create-react-app .”
- Creation of the server component
 - Browser
 - Usage of JSON Formatter Plugin
 - Go to server directory
 - npm init --yes
 - npm install -g json-server
 - Update to package.json
 - Verification
 - <http://localhost:4000/expenses>
 - <http://localhost:4000/expenses/3>
 - <http://localhost:4000/expenses?payeeName=Rahul>
- Creation of Home / App component - ExpenseTrackerApp
 - Changes
 - Remove the content from App component
 - Integrate ExpenseTrackerApp to App component
- API to retrieve all the expense items
 - npm install axios
 - Service.js
- Fetching the expense items
 - ExpenseTrackerApp component
- Displaying the expense items in a Table format
 - New Component
 - ExpenseltemsLister.js
 - Install react-bootstrap libraries
 - Reference table from react-bootstrap
 - <https://react-bootstrap.netlify.app/docs/components/table/>
 - Columns
 - ID
 - Expense Description
 - Payee
 - Date

- Amount
- ExpenseTrackerApp
 - Represent expenseltems through state
 - Include the ExpenseltemsLister component
 - Pass expenseltems [state] to ExpenseltemsLister
- Expense Addition - UI - Part-I
 - Creation of a new Component
 - ExpenseCreator.js / ExpenseCreation.js
 - Bootstrap Reference
 - -
 - <https://react-bootstrap.netlify.app/docs/components/modal/#live-demo>
 - Add a Button to this component
 - Add the supporting Modal component
- Expense Addition - Part II
 - Modal Box
 - Change Title
 - Modal Body
 - Create a Form
 - Have 4 fields
 - Description
 -
 - Payee
 - Amount
 - Date
 - <https://react-bootstrap.netlify.app/docs/forms/overview/>
 - 4 Fields
 - Description
 - Text
 - Payee
 - Bring the initial version
 - <https://react-bootstrap.netlify.app/docs/forms/select/#formselect>
 - Price
 - Number
 - Expense Date
 - Date
- Expense Addition - Part III
 - Create a new file - expense-utils.js
 - getAllPayeeNames()
 - ExpenseCreator
 - payeeNames
 - OPTION [1]
 - Select a Payee Name
 - Dynamically generate the options
 - Value = {payeeName}
 - textContent - {payeeName}
- Saving Data - Part I
 - Post API to save the data

- Saving Data - Part II
 - Add onSubmit support at the Form level
 - Have a event handler implementation
 - handleSubmit
 - Collect all 4 user supplied values
 - useRef
 - Call the service-API post method
 - Input
 - ['movie', 'Rahul', '12/12/2023', '600']
 - Output
 - ['movie', 'Rahul', '12/12/2023', '600', id : 'NEW_CONTENT']
 - Print the response in the console
- Summary Table
 - Create a new Component
 - ExpenseSummary
 - Methods
 - getGrandTotal()
 - getTotalExpenseByPayee(payee)
 - getPendingAmount(payee)
 - Table
 - Integrate the new component ExpenseTrackerApp