Full Stack Developer

1. Order Stats Widget

Create a react application "**OrderStatsApp**" using typescript/scss. App shows the order stats widget similar to the design "widget_order_stats@2x.png". Data of the statistics should come from json file "data.json".

Create a component "WidgetOrderStats" accepts the data and display all the stats. Use this component in the application.

Also this component should be responsive. This widget component uses a sub-component "OrderStat" to display each statistic.

2. Bank Account Simulation

Create a typescript based api “**BackAccount**" class with following functionalities:

- openAccount - used to start the account. Parameters passed are "name, gender, dob, email, mobile, address, initialBalance, adharNo, panNo, etc". Store the information as account details. Only after opening we can do transactions on this account.
- updateKYC - used to update the account KYC details - Parameters passed are "name, dob, email, mobile, adharNo, panNo". Update the KYC information in account details
- depositMoney - used to deposit money in the account. Parameters passed are "amount". Update the balance and create the ledger for this transaction
- withdrawMoney - used to withdraw money in the account. Parameters passed are "amount". Update the balance and create the ledger for this transaction
- transferMoney - used to transfer money to another account. Parameters passed are "toName, amount". Update the balance and create the ledger for this transaction
- receiveMoney - used to receive money to another account. Parameters passed are "fromName, amount". Update the balance and create the ledger for this transaction

- printStatement - Print the account details and transaction statements in neat format.
- closeAccount - Used to close the account. After closing the account no transactions can be done on this account

3. Responsive course tables

Create a react application "**CourseApp**" using typescript/scss. App shows the responsive course table similar to design "courses_table@2x.jpg".

Create a server node js application "**CourseServer**" and return the data of the courses. App should use the data from server and display it

4. Simple Calc

Create a node js command-line program "**Calc**" which accepts a question as string eg "What is 5 plus 7?". This program parses the question and calculate the result and print it eg: "5 plus 7 is 12".

Program should support for add, subtract, multiply and divide operations.

Input arg: What is 5 plus 7?

Output: 5 plus 7 is 12

Input arg: What is 9 minus 3?

Output: 9 minus 3 is 6

Input arg: What is 24 multiplied by 2?

Output: 24 multiplied by 2 is 48

Input arg: What is 60 divided by 5?

Output: 60 divided by 5 is 12

5. Responsive home screen

Create a html/scss program to display responsive home screen similar to the design "home_screen@2x.jpg"

6. Simple User API

Create a node js express based program "**UserApiServer**" which provides following REST Apis:

- GET /users - Used to return list of the users which information like "id, name, createdOn, gender, dob, city, state, pincode, modifiedOn"
- POST /users - Used to create new user
- PUT /users/<userId> - Used to update an existing user
- DELETE /users/<userId> - Used to delete an existing user

Note: You can store the users information in memory or json data file

7. Responsive transactions

Create a responsive application "**TransactionsApp**" using typescript/scss. App shows transactions list similar to design "transactions@2x.jpg".

Create a component "TransactionList" to display the transactions list. This component should be fully responsive.