REST APIS (Intermediate)

- Write a schema in your node js application to create a new collection with the name User.
- For the schema, initialize the below fields :
    - **firstName** : String
    - **lastName** : String
    - **emailId** : String
    - **phoneNumber** : Number
    - **education** : {
        “primaryEducation” : String,
        “secondaryEducation” : String,
        “tertiaryEducation” : String
      }
    - **hobby** : [
        {
          “type” : String (indoor hobby),
          “value” : String (Name of the hobby)
        },
        {
          “type” : String (outdoor hobby),
          “value” : String (Name of the hobby)
        }
      ]
      [hobby field should have two elements in the array one for indoor and other for outdoor. Don’t accept more than two elements]

[Note : emailId and phoneNumber should be unique for every document/student within a collection (This is also called indexing)]

- Now create the below APIs in your node js application to **CREATE**, **READ**, **UPDATE** and **DELETE** users/documents from the collections
  - **GET : /user/get** -
    - To get all the users from the collection.
    - Traverse through the collection and return all the documents.
    - Response (200) : {

```
"message" : "success",
"data" : [
  {
    "firstName" : "Rahul",
    "lastName" : "Sharma"
    "emailId" : "abc@gmail.com",
    "phoneNumber" : 954235556,
    "education" : {
        "primaryEducation" : "St. Mary's Public
                              School",
        "secondaryEducation" : "Standard High
                               School",
        "tertiaryEducation" : "Govt. College of
                              Eng"
    },
    "hobby" : [
        {
            "type" : "indoor",
            "value" : "chess"
```

```
        },
        {
            “type” : “outdoor”,
            “value” : “swimming”
        }
    ]
},
{
    “firstName” : “Yash”,
    “lastName” : “Gupta”
    “emailId” : “ajjc@gmail.com”,
    “phoneNumber” : 954235556,
    “education” : {
        “primaryEducation” : “St. Antony’s
                                Public School”,
        “secondaryEducation” : “Standard High
                                School”,
        “tertiaryEducation” : “ABC College of
                                Eng”
    },
    “hobby” : [
        {
            “type” : “indoor”,
            “value” : “books”
        },
        {
            “type” : “outdoor”,
            “value” : “badminton”
        }
    ]
```

```
        ]
    }
```

- **GET : /user/get?emailId=abc@gmail.com**
  - To get the information of a student with the given emailId.
  - Run a query with the given emailId to match the document and then return that document.
  - Before running the query, do a validation check on the emailId (Check if the emailId is a valid string or not)
  - Response (200) : {

```
"message" : "success",
"data" :
 {
    "firstName" : "Rahul",
    "lastName" : "Sharma"
    "emailId" : "abc@gmail.com",
    "phoneNumber" : 954235556,
    "education" : {
        "primaryEducation" : "St. Mary's Public
                              School",
        "secondaryEducation" : "Standard High
                               School",
        "tertiaryEducation" : "Govt. College of
                              Eng"
    },
    "hobby" : [
        {
```

```
                    "type" : "indoor",
                    "value" : "chess"
                },
                {
                    "type" : "outdoor",
                    "value" : "swimming"
                }
            ]
        }
    }
```

- **GET : /user/get?phoneNumber=657895233**
  - To get the information of a student with the given phoneNumber.
  - Run a query with the given emailId to match the document and then return that document.
  - Before running the query, do a validation check on the phoneNumber.
  - Response (200) : {

```
"message" : "success",
"data" :
 {
    "firstName" : "Rahul",
    "lastName" : "Sharma"
    "emailId" : "abc@gmail.com",
    "phoneNumber" : 657895233,
    "education" : {
        "primaryEducation" : "St. Mary's Public
                              School",
        "secondaryEducation" : "Standard High
                                School",
```

```
            "tertiaryEducation" : "Govt. College of
                                   Eng"
        },
        "hobby" : [
            {
                "type" : "indoor",
                "value" : "chess"
            },
            {
                "type" : "outdoor",
                "value" : "swimming"
            }
        ]
    }
}
```


- **POST : /user/create** -
  - To create a new document in the CSE collection with student information.
  - Take the request body and feed it in the collection.
  - Before running the query, do a validation check on the firstName, lastName,emailId, phoneNumber, education and hobby (Check if the fields are valid or not)
  - Also check that if the emailId/phoneNumber already exists in the collection, if not then return a 400 error with a proper message.
  - Request Body :


```
{
```

```
"firstName" : "Rahul",
"lastName" : "Sharma"
"emailId" : "abc@gmail.com",
"phoneNumber" : 954235556,
"education" : {
    "primaryEducation" : "St. Mary's Public
                          School",
    "secondaryEducation" : "Standard High
                           School",
    "tertiaryEducation" : "Govt. College of
                          Eng"
},
"hobby" : [
    {
        "type" : "indoor",
        "value" : "chess"
    },
    {
        "type" : "outdoor",
        "value" : "swimming"
    }
]
}.
```

- Response (200):

```
{
  "message" : "success",
  "data" : {
        "firstName" : "Rahul",
```

```
        "lastName" : "Sharma"
        "emailId" : "abc@gmail.com",
        "phoneNumber" : 954235556,
        "education" : {
            "primaryEducation" : "St. Mary's Public
                                  School",
            "secondaryEducation" : "Standard High
                                    School",
            "tertiaryEducation" : "Govt. College of
                                   Eng"
        },
        "hobby" : [
            {
                "type" : "indoor",
                "value" : "chess"
            },
            {
                "type" : "outdoor",
                "value" : "swimming"
            }
        ]
    }
}
```

- **POST : /user/update/phoneNumber -**
  - To update a document in the CSE collection with new information for a given emailId.
  - Run a query with the given emailId, match the document and then update the field given in the request body.

- Before running the query, do a validation check on the emailId (Check if the emailId is a valid string or not).
- Request Body :

```
{
    “emailId” : “abc@gmail.com”,
    “phoneNumber” : 954235555,
}.
```

- Response (200) :

```
{
  “message” : “success”,
  “data” :{
          “firstName” : “Rahul”,
          “lastName” : “Sharma”
          “emailId” : “abc@gmail.com”,
          “phoneNumber” : 954235555,
          “education” : {
              “primaryEducation” : “St. Mary’s Public
                                    School”,
              “secondaryEducation” : “Standard High
                                      School”,
              “tertiaryEducation” : “Govt. College of
                                    Eng”
           },
           “hobby” : [
               {
                   “type” : “indoor”,
                   “value” : “chess”
```

```
            },
            {
                "type" : "outdoor",
                "value" : "swimming"
            }
        ]
    }
}
```

- **POST : /user/update/education -**
  - To update a document in the CSE collection with new information for a given emailId.
  - Run a query with the given emailId, match the document and then update the field given in the request body.
  - Before running the query, do a validation check on the emailId and the other field.
  - Request Body :

```
{
    "emailId" : "abc@gmail.com",
    "secondaryEducation" : "FRHJ High School",
}
```

[Here it can be primaryEducation, secondaryEducation or tertiaryEducation. Update according to the field given in the request body]

  - Response (200) :

```
{
  "message" : "success",
  "data" :{
          "firstName" : "Rahul",
```

```
        "lastName" : "Sharma"
        "emailId" : "abc@gmail.com",
        "phoneNumber" : 954235555,
        "education" : {
            "primaryEducation" : "St. Mary's Public
                                  School",
            "secondaryEducation" : "Standard High
                                    School",
            "tertiaryEducation" : "Govt. College of
                                   Eng"
        },
        "hobby" : [
            {
                "type" : "indoor",
                "value" : "chess"
            },
            {
                "type" : "outdoor",
                "value" : "swimming"
            }
        ]
    }
}
```


- **POST : /user/update/hobby -**
  - To update a document in the CSE collection with new information for a given emailId.
  - Run a query with the given emailId, match the document and then update the field given in the request body.

- Before running the query, do a validation check on the emailId and hobby.
- Request Body :

```
{
    “emailId” : “abc@gmail.com”,
    “hobby” : {
        “type” : “indoor”
        “value” : “dance”
    },
}
```

[Here it can be primaryEducation, secondaryEducation or tertiaryEducation. Update according to the field given in the request body]

- Response (200) :

```
{
  “message” : “success”,
  “data” :{
          “firstName” : “Rahul”,
          “lastName” : “Sharma”
          “emailId” : “abc@gmail.com”,
          “phoneNumber” : 954235555,
          “education” : {
              “primaryEducation” : “St. Mary’s Public
                                    School”,
              “secondaryEducation” : “Standard High
                                      School”,
              “tertiaryEducation” : “Govt. College of
                                     Eng”
          },
```

```
            "hobby" : [
                {
                    "type" : "indoor",
                    "value" : "dance"
                },
                {
                    "type" : "outdoor",
                    "value" : "swimming"
                }
            ]
        }
    }
```

- **DELETE : /user/delete?emailId=abc@gmail.com -**
  - To delete a document in the CSE collection for a given emailId.
  - Run a query with the given emailId and delete the matched document.
  - Before running the query, do a validation check on the emailId (Check if the emailId is a valid string or not).
  - Response (200) : {
    "message" : "Document deleted successfully"
    }
- **DELETE : /user/delete?phoneNumber=456263335 -**
  - To delete a document in the CSE collection for a given phoneNumber.
  - Run a query with the given phoneNumber and delete the matched document.

        - Before running the query, do a validation check on the phoneNumber.
        - Response (200) : {
          “message” : “Document deleted successfully”
          }

- For all the above **APIs**, do return the error responses :
    - **Check for validations** :
        - Check if any of the fields in the request are invalid, if they are then return a **400** status code along with the following message :
            - Response (400) :
              {
                “message” : “Missing mandatory parameters”
              }
            - Missing mandatory parameters means that the parameters/fields required by the particular API are missing.
            - Change the message according to the API requirement.
    - **Check for exceptions/runtime errors and catch them :**
        - Wrap the code in try-catch method to catch any of the exceptions that could occur and return that error/exception with **500** status code
            - Response (500) :
              {
                “message” : “Internal Server Error”
              }

- Internal Server error means that there was an issue encountered while the server was returning the response.