**Database Design**

User Table

* id (int PrimaryKey)
* name (Varchar)
* email (Varchar)
* password (Varchar)

There will be data dump of few user.

User Specific Table - One per user

Table name will same as **email**, unique for all the user

* id (int PrimaryKey)
* firstname (Varchar)
* lastname (Varchar)
* password (bigint)

**API Documentation**

1) Validating Server

GET <http://localhost:5000/me>

Response –

Hello Welcome to CRM App!

2) User Authentication

POST <http://localhost:5000/user>

Request –

{

    "email":"meet@gmail.com",

    "name":"Meet",

    "password":"Meet"

}

Response –

{

    "msg": [

        {

            "id": 1,

            "name": "Meet",

            "email": "meet@gmail.com",

            "password": "Meet"

        }

    ]

}

3) Create Table

POST <http://localhost:5000/createtable>

Table will be created with default columns if not created already.

Request –

{

    "name":"Meet",

    "email":"meet@gmail.com"

}

Response – msg will be based on table, it may be different for different user

{

    "msg": "Table exists"

}

4) Get Table

GET <http://localhost:5000/gettable/:tablename>

Response – msg

{

    "msg": {

        "columns": {

            "0": "id",

            "1": "firstname",

            "2": "lastname",

            "3": "phone"

        },

        "tables": [

            {

                "id": 3,

                "firstname": "Het",

                "lastname": "Parekh",

                "phone": 95998564578

            },

            {

                "id": 4,

                "firstname": "Sandy",

                "lastname": "Sahrma",

                "phone": 7825984555

            }

        ]

    }

}

5) Insert Row

POST <http://localhost:5000/insertrow>

Row will be added to the table by User

Request –

{

    "email":"jainam@gmail.com",

    "columnName":{

        "firstname":"Arush",

        "lastname":"Samay",

        "phone":787946141

    }

}

Response – It will be based on query result

{

    "msg": {

        "fieldCount": 0,

        "affectedRows": 1,

        "insertId": 2,

        "serverStatus": 2,

        "warningCount": 0,

        "message": "",

        "protocol41": **true**,

        "changedRows": 0

    }

}

6) Add Column

PUT <http://localhost:5000/addcolumn>

Request – isText use to define if column is number or varchar type

{

    "email":"meet@gmail.com",

    "name":"Meet",

    "columnName":"height",

    "isText":**false**

}

Response – msg will be based on table, it may be different for different user

{

    "msg": {

        "fieldCount": 0,

        "affectedRows": 0,

        "insertId": 0,

        "serverStatus": 2,

        "warningCount": 0,

        "message": "&Records: 0  Duplicates: 0  Warnings: 0",

        "protocol41": **true**,

        "changedRows": 0

    }

}

7) Update Column

PUT <http://localhost:5000/updatecolumn>

Request –

{

    "email":"meet@gmail.com",

    "id":"3",

    "columns": {

        "firstname":"Bittu",

        "lastname":"Paul"

   }

}

Response – msg will be based on table, it may be different for different user

{

    "msg": "Tabel Updated Successfully"

}