**INSTRUCTIONS TO USE API**

1. For staff data:
   1. POST or add new staff –

Send POST request to <http://localhost:3000/api/staff>

adding data format => Staff Name (form name attribute is “name”),

Staff Email (form name attribute is “email”),

Staff Password (form name attribute is “password”),

Staff Type (form name attribute is “type”)

* 1. PUT or update staff details –

Send PUT request to <http://localhost:3000/api/staff>

Data format => same as above + Staff ID (attribute is”id”)

* 1. GET all staff details

Send GET request to <http://localhost:3000/api/staff>

Return json data according to mysql table format

* 1. GET one staff details

Send GET request to <http://localhost:3000/api/staff/:>email

eg. [http://localhost:3000/api/staff/](http://localhost:3000/api/staff/1)abc@abc.com

returns json data

* 1. DELETE one staff

Send DELETE request to <http://localhost:3000/api/staff/:email>

Eg. <http://localhost:3000/api/staff/abc@abc.com>

1. For customer data:
   1. POST or add new customer –

Send POST request to [http://localhost:3000/api/](http://localhost:3000/api/staff)customer

adding data format => Customer Name (form name attribute is “name”),

Customer Email (form name attribute is “email”),

Customer Password (form name attribute is “password”)

* 1. PUT or update customer details –

Send PUT request to [http://localhost:3000/api/](http://localhost:3000/api/staff)customer

Data format => same as above + Customer ID (attribute is”id”)

* 1. GET all customer details

Send GET request to [http://localhost:3000/api/](http://localhost:3000/api/staff)customer

Return json data according to mysql table format

* 1. GET one customer details

Send GET request to <http://localhost:3000/api/customer/:>email

eg. [http://localhost:3000/api/customer/abc@abc.com](http://localhost:3000/api/customer/1)

returns json data

* 1. DELETE one customer

Send DELETE request to <http://localhost:3000/api/customer/:email>

Eg. <http://localhost:3000/api/customer/abc@abc.com>

1. For table data:
   1. POST or add new table –

Send POST request to [http://localhost:3000/api/](http://localhost:3000/api/staff)table

adding data format => Table Type (form name attribute is “type”),

Availability (form name attribute is “availability”, value is ‘yes’ or ‘no’)

* 1. PUT or update table details –

Send PUT request to [http://localhost:3000/api/](http://localhost:3000/api/staff)table

Data format => same as above + Table ID (attribute is”id”)

* 1. GET all table details

Send GET request to [http://localhost:3000/api/](http://localhost:3000/api/staff)table

Return json data according to mysql table format

* 1. GET one table details

Send GET request to <http://localhost:3000/api/table/:id>

eg. <http://localhost:3000/api/table/1>

returns json data

* 1. DELETE one table

Send DELETE request to [http://localhost:3000/api/table/:](http://localhost:3000/api/table/:email)id

Eg. <http://localhost:3000/api/table/>1

1. For reservations data:
   1. POST or add new reservation –

Send POST request to [http://localhost:3000/api/](http://localhost:3000/api/staff)reservation

adding data format => Customer ID (form name attribute is “custID”),

Table ID (form name attribute is “tableID”),

Date (form name attribute is “date”),

Time (form name attribute is “time”)

* 1. PUT or update reservation details –

Send PUT request to [http://localhost:3000/api/](http://localhost:3000/api/staff)reservation

Data format => same as above + Reservation ID (attribute is”id”) and minus Customer ID

* 1. GET all reservation details

Send GET request to [http://localhost:3000/api/](http://localhost:3000/api/staff)reservation

Return json data according to mysql table format

* 1. GET one reservation details

Send GET request to <http://localhost:3000/api/reservation/:id>

eg. <http://localhost:3000/api/reservation/1>

id is customer ID. To get reservations of particular customer.

returns json data

* 1. DELETE one reservation

Send DELETE request to <http://localhost:3000/api/reservation/:>id

Eg. <http://localhost:3000/api/reservation/1>

Id is reservation ID.

**INSTRUCTIONS TO RUN THE SERVER ON LOCALHOST**

1. Install node.js from <https://nodejs.org/>
2. After installing make sure its working by running following commands from Terminal –
   1. “node –v”
   2. “npm –v”
3. Change directory (cd) into the backend folder
4. Run the command “npm install” from terminal
5. When that is done run “npm start” from terminal
6. Server is now running at localhost:3000
7. Also run MySQL in phpmyadmin using WAMP or XAMPP