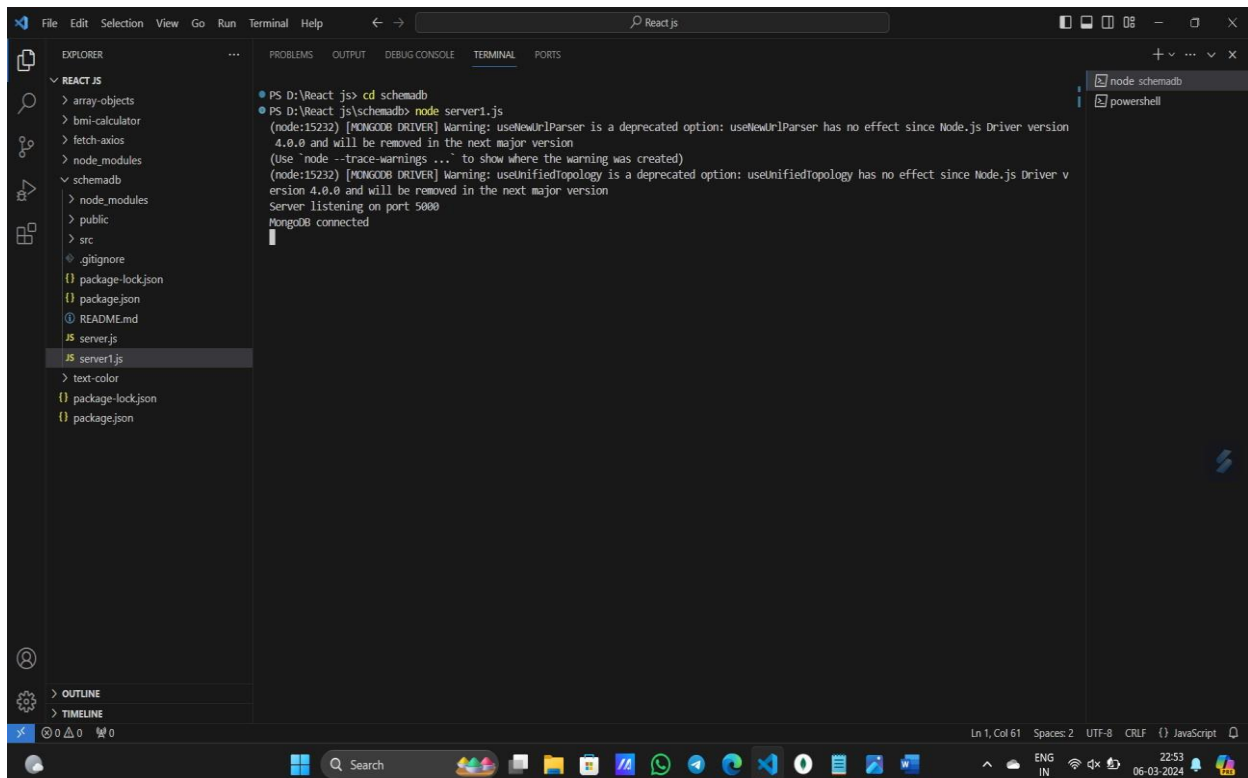


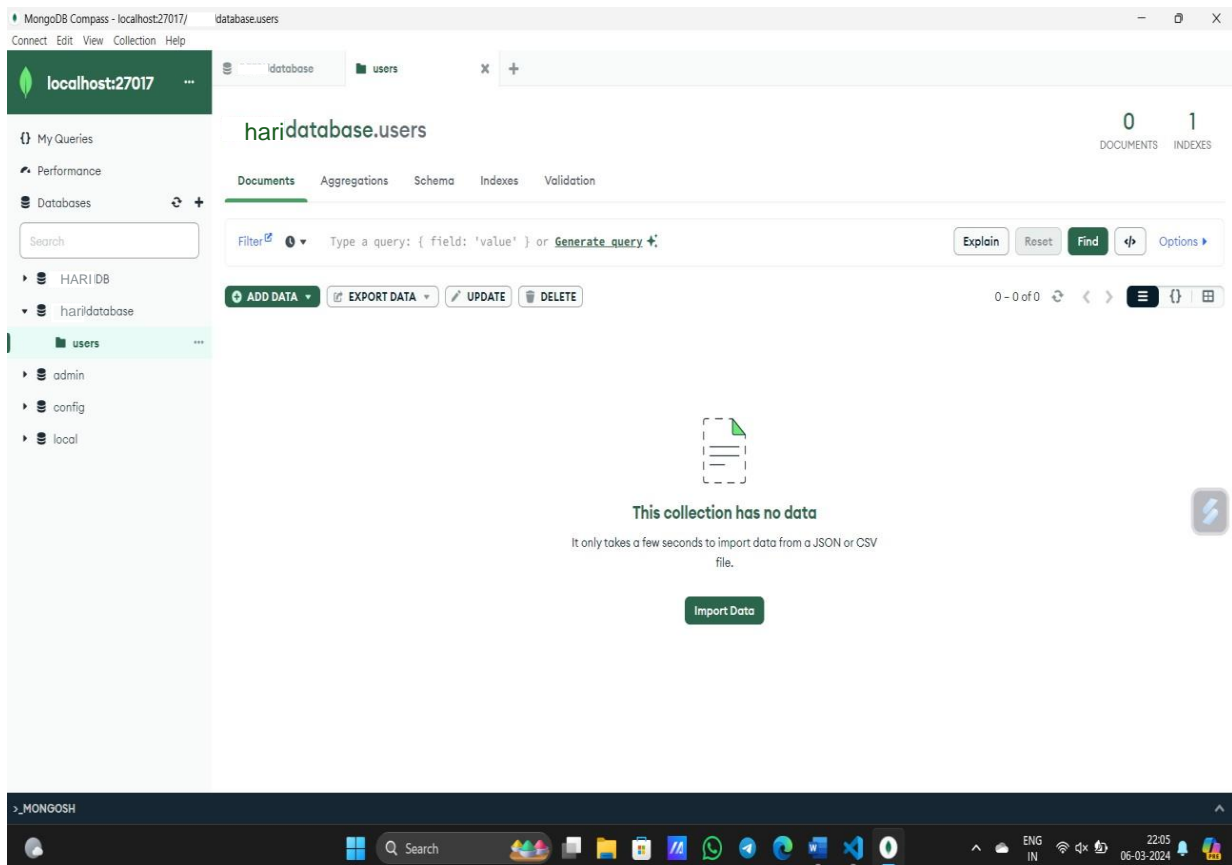
OUTPUT:

Connection to MongoDB:



The screenshot shows the VS Code interface with the 'TERMINAL' tab active. The terminal output shows the following commands and responses:

```
PS D:\React js> cd schemadb
PS D:\React js\schemadb> node server1.js
(node:15232) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:15232) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port 5000
MongoDB connected
```



Inserting user documents into MongoDB database:

The screenshot shows the VS Code interface with the Explorer, Terminal, and REST client tabs. The Explorer shows a project structure with a 'server.js' file. The Terminal shows the execution of REST client requests to insert user documents into a MongoDB database. The REST client shows the response for each request, including the user's name, email, age, and a unique ID.

```

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Anusha", "email": "anusha@example.com", "age": 20}'

name : Anusha
email : anusha@example.com
age   : 20
_id   : 65e8a6c98226ab35690e7272
_v    : 0

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Jhansi", "email": "jansi@example.com", "age": 22}'

name : Jhansi
email : jansi@example.com
age   : 22
_id   : 65e8a6e8226ab35690e7274
_v    : 0

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Deepthi", "email": "deepthi@example.com", "age": 19}'

name : Deepthi
email : deepthi@example.com
age   : 19
_id   : 65e8a7158226ab35690e7276
_v    : 0

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Vani", "email": "vani@example.com", "age": 24}'
  
```

This screenshot continues the sequence of REST client requests from the previous one. It shows the insertion of 'Vani' and 'Mrudula' into the MongoDB database. The responses show the same fields as before: name, email, age, _id, and _v.

```

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Deepthi", "email": "deepthi@example.com", "age": 19}'

name : Deepthi
email : deepthi@example.com
age   : 19
_id   : 65e8a7158226ab35690e7276
_v    : 0

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Vani", "email": "vani@example.com", "age": 24}'

name : Vani
email : vani@example.com
age   : 24
_id   : 65e8a7328226ab35690e7278
_v    : 0

PS D:\React js> Invoke-RestMethod -Method Post -Uri "http://localhost:5000/api/users" -ContentType "application/json" -Body '{"name": "Mrudula", "email": "mrudula@example.com", "age": 18}'

name : Mrudula
email : mrudula@example.com
age   : 18
_id   : 65e8a7598226ab35690e727a
_v    : 0

PS D:\React js>
  
```

MongoDB Compass - localhost:27017 / database.users

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases

Search

HARI DB

hari database

users

admin

config

local

haridatabase.users

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{ "_id": ObjectId("65e8a6c98226ab35699e7272"), "name": "Anusha", "email": "anu@example.com", "age": 28, "__v": 0 }
```

```
{ "_id": ObjectId("65e8a6eb8226ab35699e7274"), "name": "Jhansi", "email": "jaan@example.com", "age": 22, "__v": 0 }
```

```
{ "_id": ObjectId("65e8a7158226ab35699e7276"), "name": "Deepthi", "email": "deepthi@example.com", "age": 19, "__v": 0 }
```

```
{ "_id": ObjectId("65e8a7328226ab35699e7278"), "name": "Vani", "email": "vani@example.com", "age": 24, "__v": 0 }
```

> MONGOSH

MongoDB Compass - localhost:27017 / database.users

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases

Search

HARI DB

hari database

users

admin

config

local

haridatabase.users

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{ "name": "Jhansi", "email": "jaan@example.com", "age": 22, "__v": 0 }
```

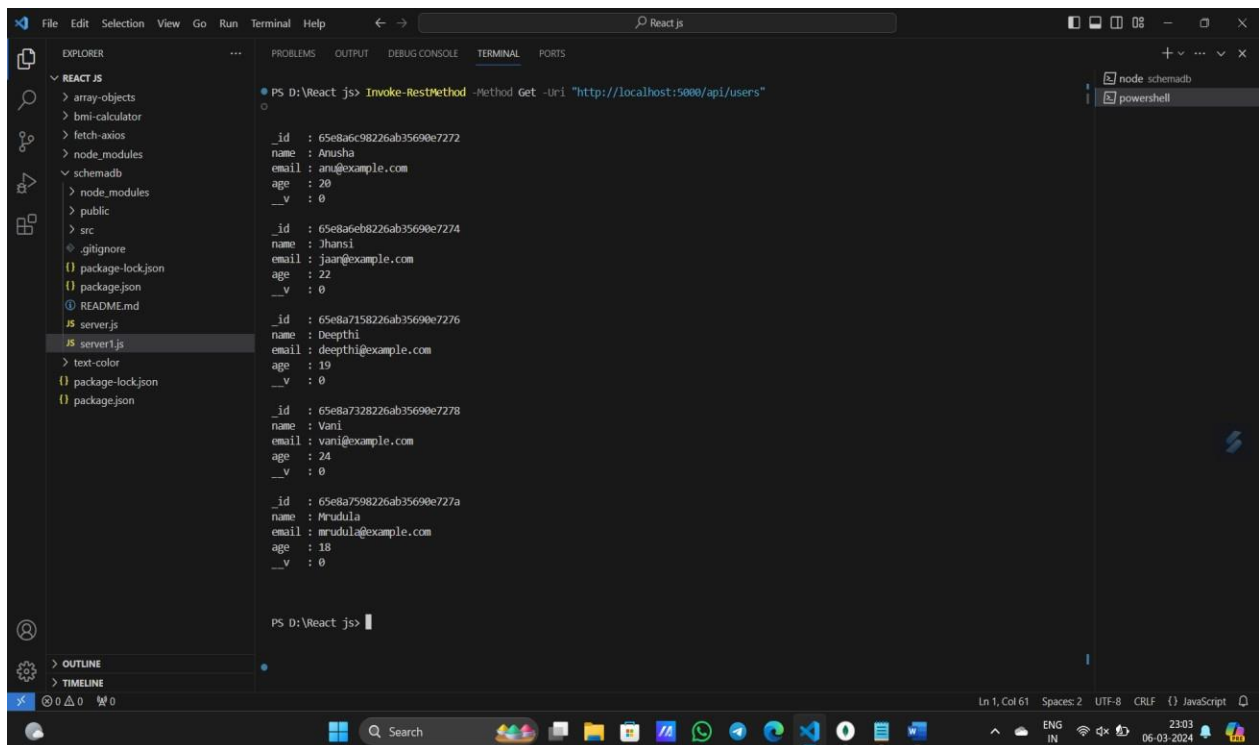
```
{ "_id": ObjectId("65e8a7158226ab35699e7276"), "name": "Deepthi", "email": "deepthi@example.com", "age": 19, "__v": 0 }
```

```
{ "_id": ObjectId("65e8a7328226ab35699e7278"), "name": "Vani", "email": "vani@example.com", "age": 24, "__v": 0 }
```

```
{ "_id": ObjectId("65e8a7598226ab35699e727a"), "name": "Mrudula", "email": "mrudula@example.com", "age": 18, "__v": 0 }
```

> MONGOSH

Get all User documents:

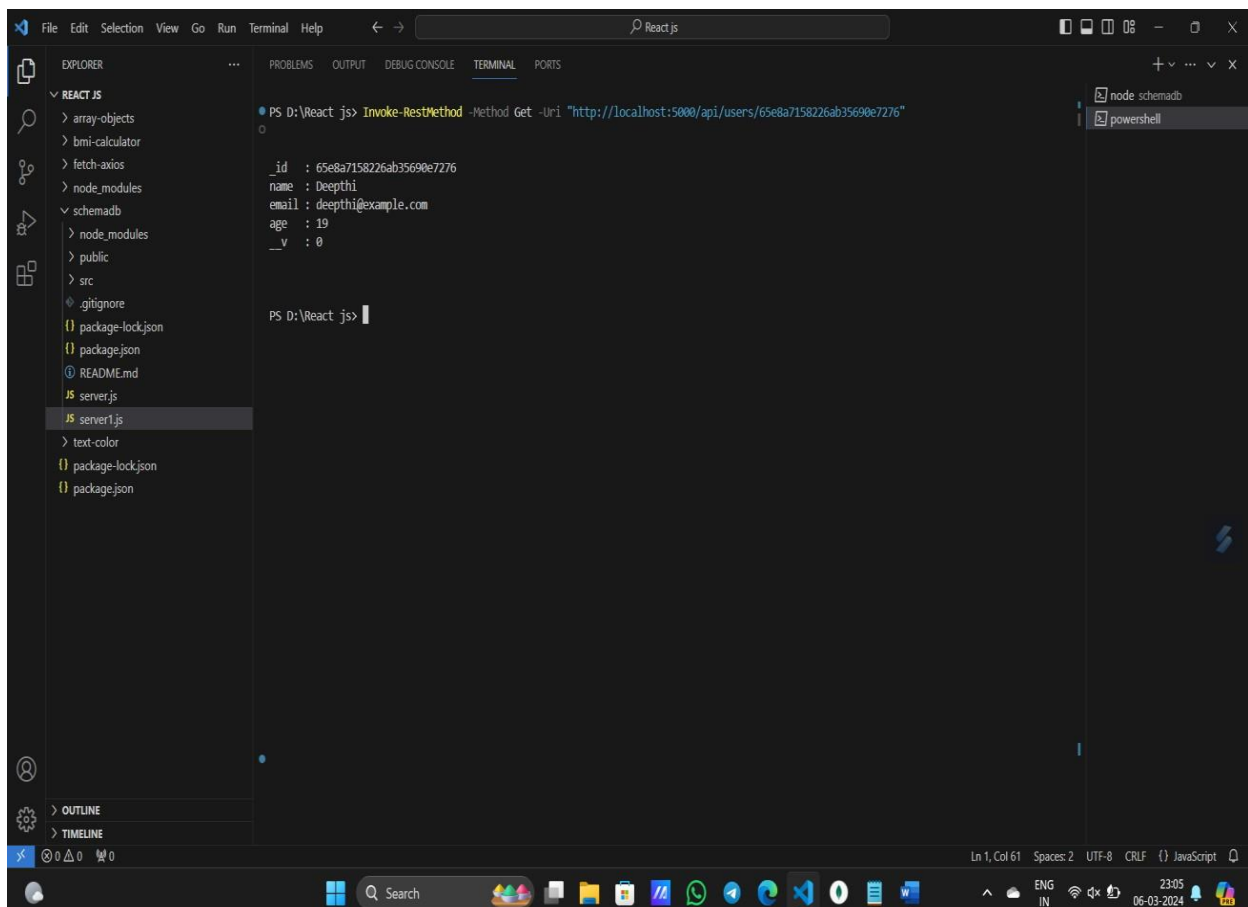


The screenshot shows the VS Code interface with the Explorer, Search, and Run and Debug panels. The Explorer panel shows the project structure with folders like array-objects, bmi-calculator, fetch-axios, node_modules, schemadb, public, src, .gitignore, package-lock.json, package.json, README.md, server.js, server1.js, text-color, package-lock.json, and package.json. The Search panel shows the results of a search for 'server.js'. The Run and Debug panel shows the output of a REST client call to 'http://localhost:5000/api/users' using the 'Invoke-RestMethod' command. The output displays a list of user documents with fields like _id, name, email, age, and __v.

```
PS D:\React js> Invoke-RestMethod -Method Get -Uri "http://localhost:5000/api/users"

{
  "_id": "65e8a6c98226ab35690e7272",
  "name": "Anusha",
  "email": "anusha@example.com",
  "age": 20,
  "__v": 0
},
{
  "_id": "65e8a6eb8226ab35690e7274",
  "name": "Jhansi",
  "email": "jaan@example.com",
  "age": 22,
  "__v": 0
},
{
  "_id": "65e8a7158226ab35690e7276",
  "name": "Deepthi",
  "email": "deepthi@example.com",
  "age": 19,
  "__v": 0
},
{
  "_id": "65e8a7328226ab35690e7278",
  "name": "Vani",
  "email": "vani@example.com",
  "age": 24,
  "__v": 0
},
{
  "_id": "65e8a7598226ab35690e727a",
  "name": "Mrudula",
  "email": "mrudula@example.com",
  "age": 18,
  "__v": 0
}
```

Retrieving specific User Document By Id:

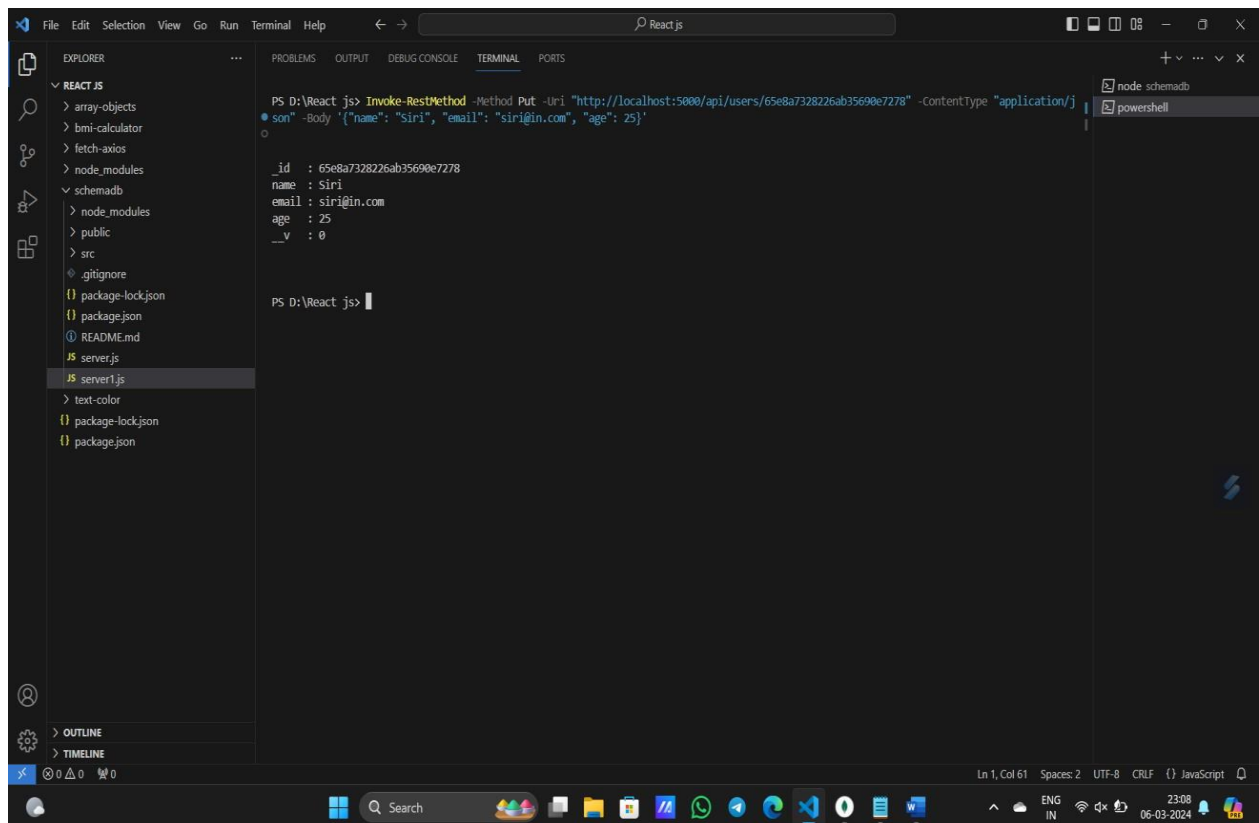


The screenshot shows the VS Code interface with the Explorer, Search, and Run and Debug panels. The Explorer panel shows the project structure with folders like array-objects, bmi-calculator, fetch-axios, node_modules, schemadb, public, src, .gitignore, package-lock.json, package.json, README.md, server.js, server1.js, text-color, package-lock.json, and package.json. The Search panel shows the results of a search for 'server1.js'. The Run and Debug panel shows the output of a REST client call to 'http://localhost:5000/api/users/65e8a7158226ab35690e7276' using the 'Invoke-RestMethod' command. The output displays a single user document with fields like _id, name, email, age, and __v.

```
PS D:\React js> Invoke-RestMethod -Method Get -Uri "http://localhost:5000/api/users/65e8a7158226ab35690e7276"

{
  "_id": "65e8a7158226ab35690e7276",
  "name": "Deepthi",
  "email": "deepthi@example.com",
  "age": 19,
  "__v": 0
}
```

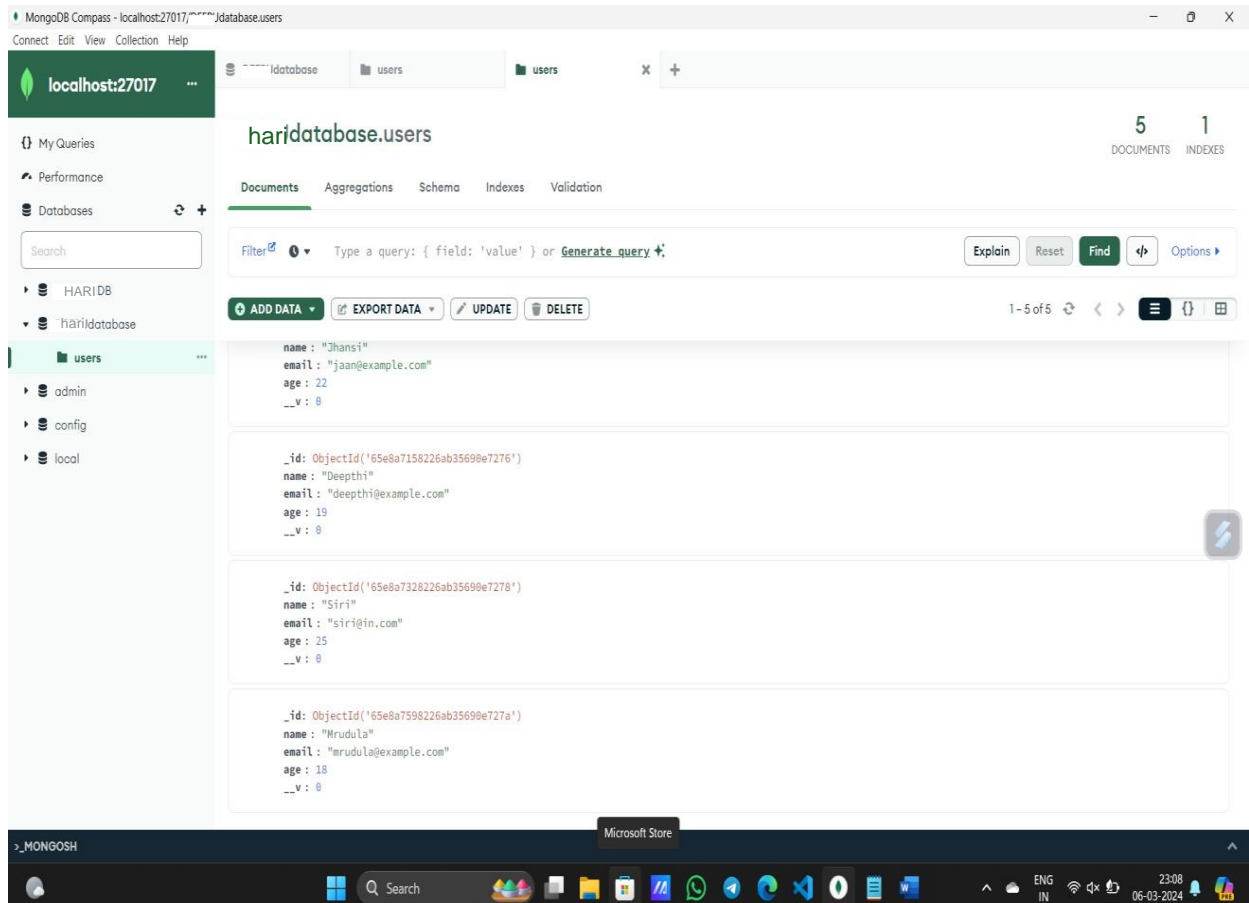
Updating Specific User Document By Id:



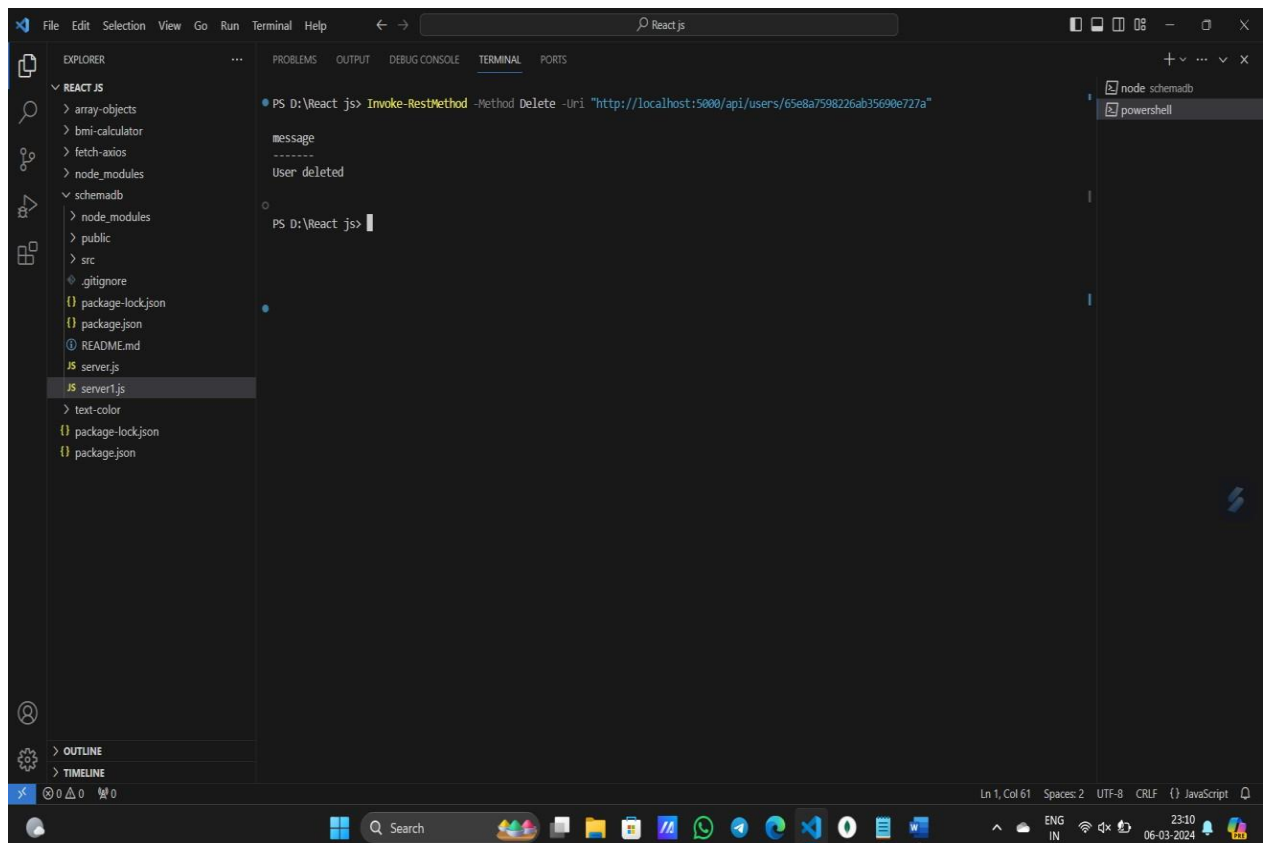
The screenshot shows the VS Code interface with the Explorer, Search, and Run and Debug panels. The Explorer panel shows the file structure of the project, including the `server.js` file. The Search panel shows the results of a search for `son` in the `server.js` file. The Run and Debug panel shows the output of the `Invoke-WebRequest` command, which is a PUT request to the `http://localhost:5000/api/users/65e8a7328226ab35690e7278` endpoint. The response is a JSON object representing the updated user document.

```
PS D:\React js> Invoke-WebRequest -Method Put -Uri "http://localhost:5000/api/users/65e8a7328226ab35690e7278" -ContentType "application/json" -Body '{"name": "Siri", "email": "siri@in.com", "age": 25}'
```

```
{
  "_id": "65e8a7328226ab35690e7278",
  "name": "Siri",
  "email": "siri@in.com",
  "age": 25,
  "__v": 0
}
```



Deleting Specific User Document By Id:

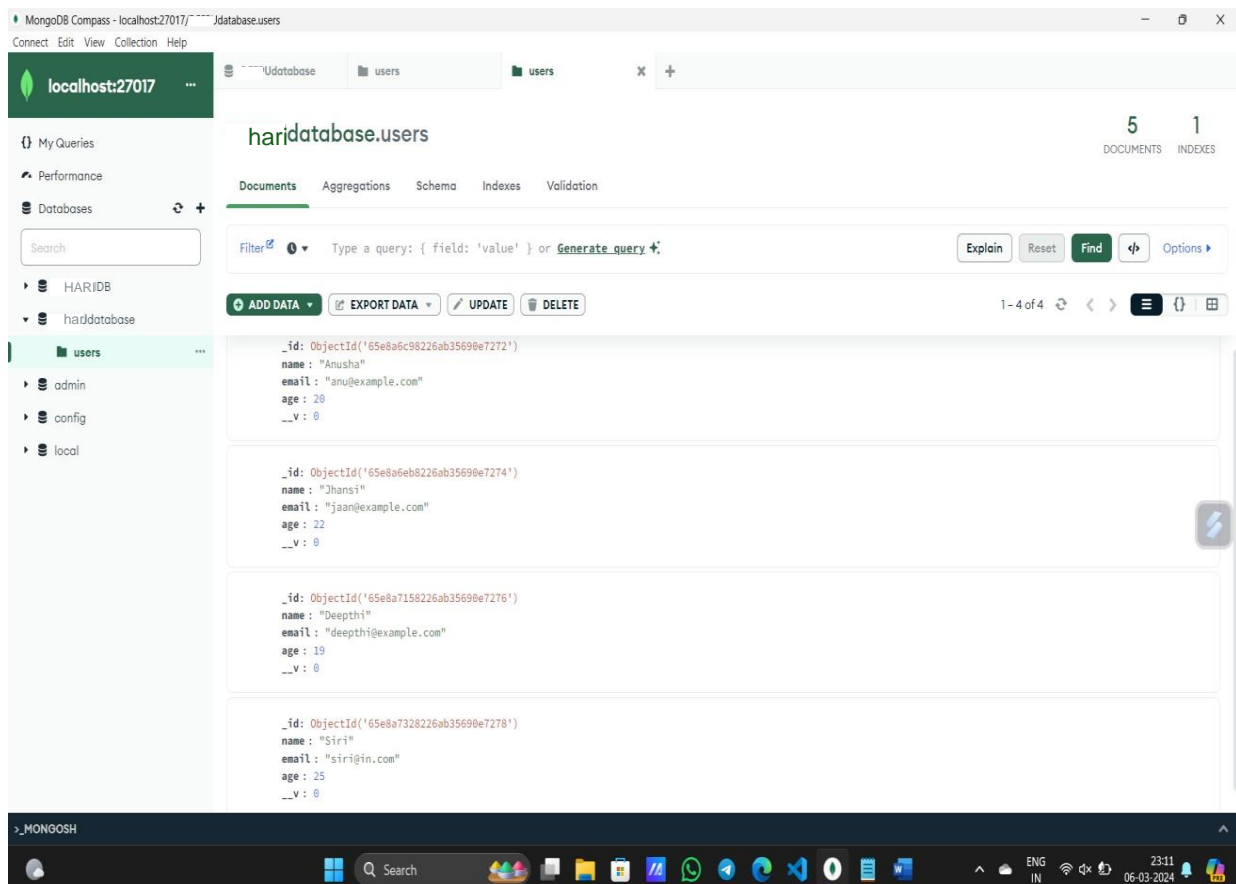


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of a project named 'React.js'. The file 'server.js' is selected. The Terminal panel on the right shows a PowerShell session where a REST client call is executed to delete a user document by ID. The output of the call is displayed in the terminal.

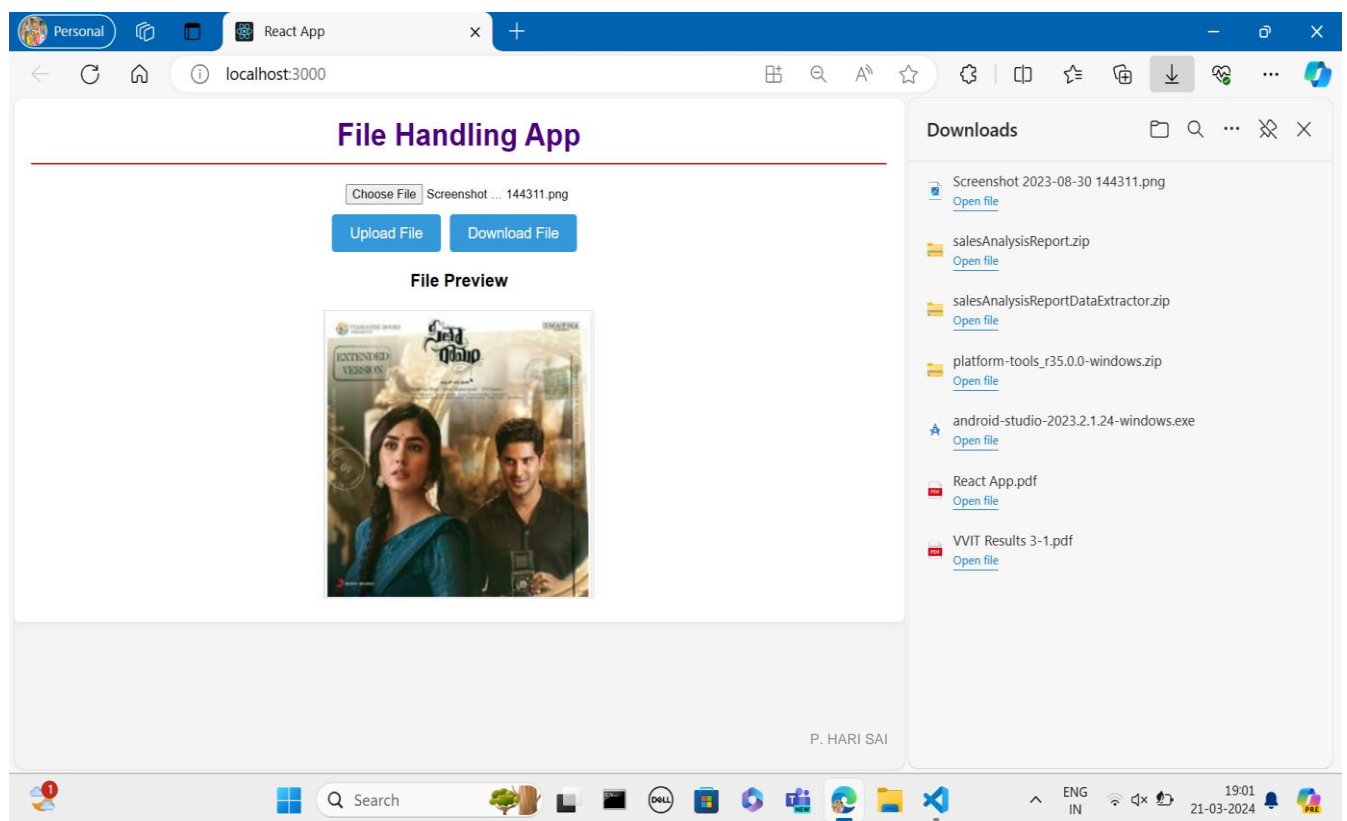
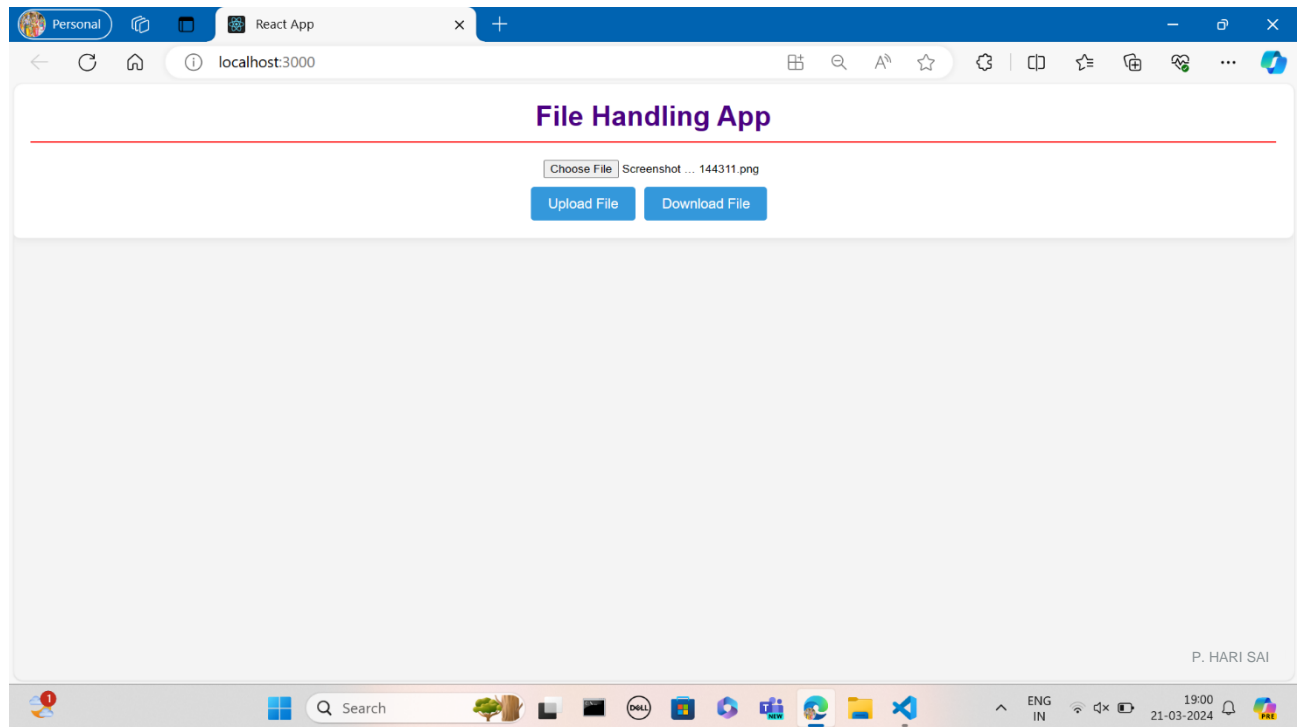
```
PS D:\React.js> Invoke-RestMethod -Method Delete -Uri "http://localhost:5000/api/users/65e8a7598226ab35690e727a"

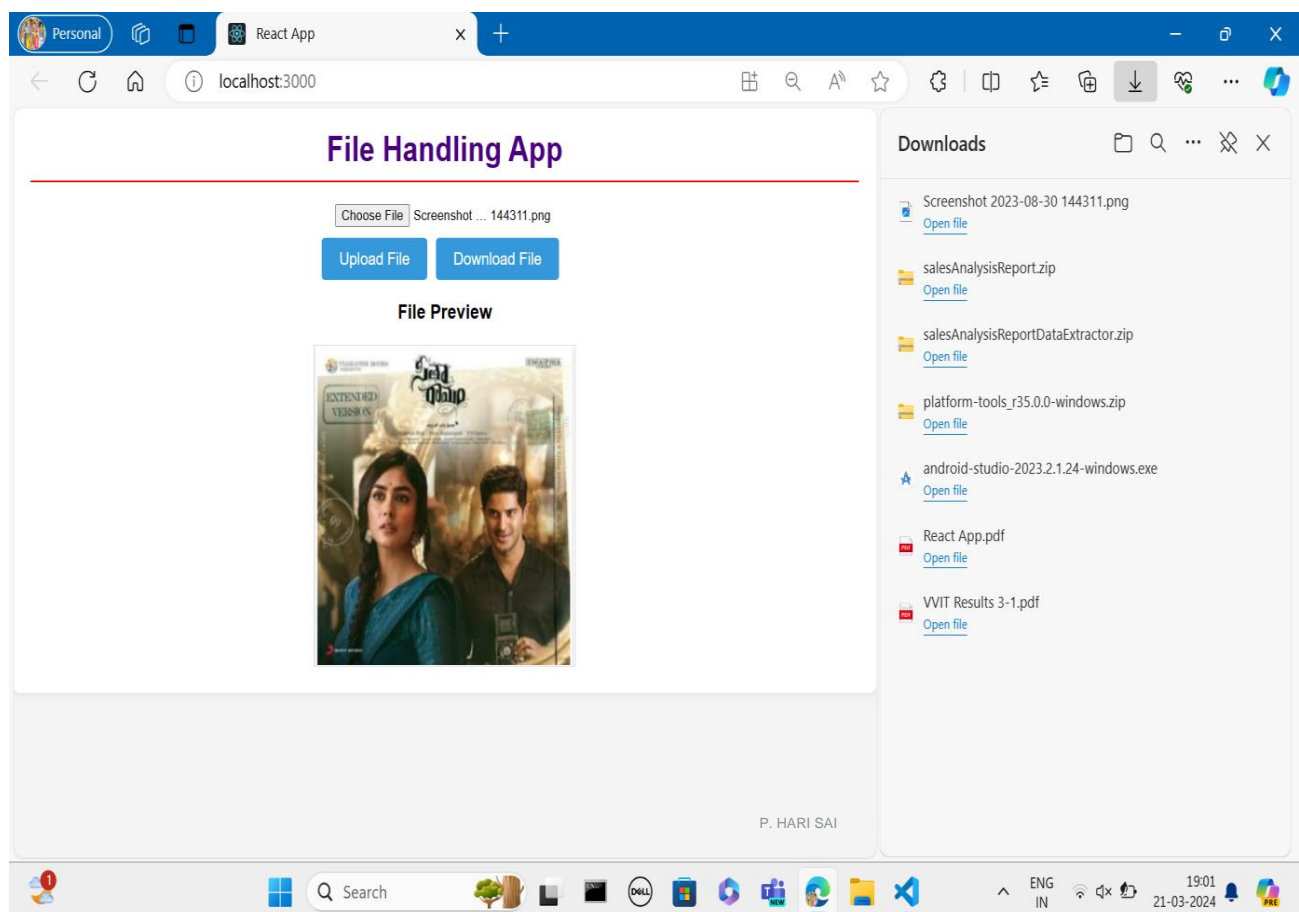
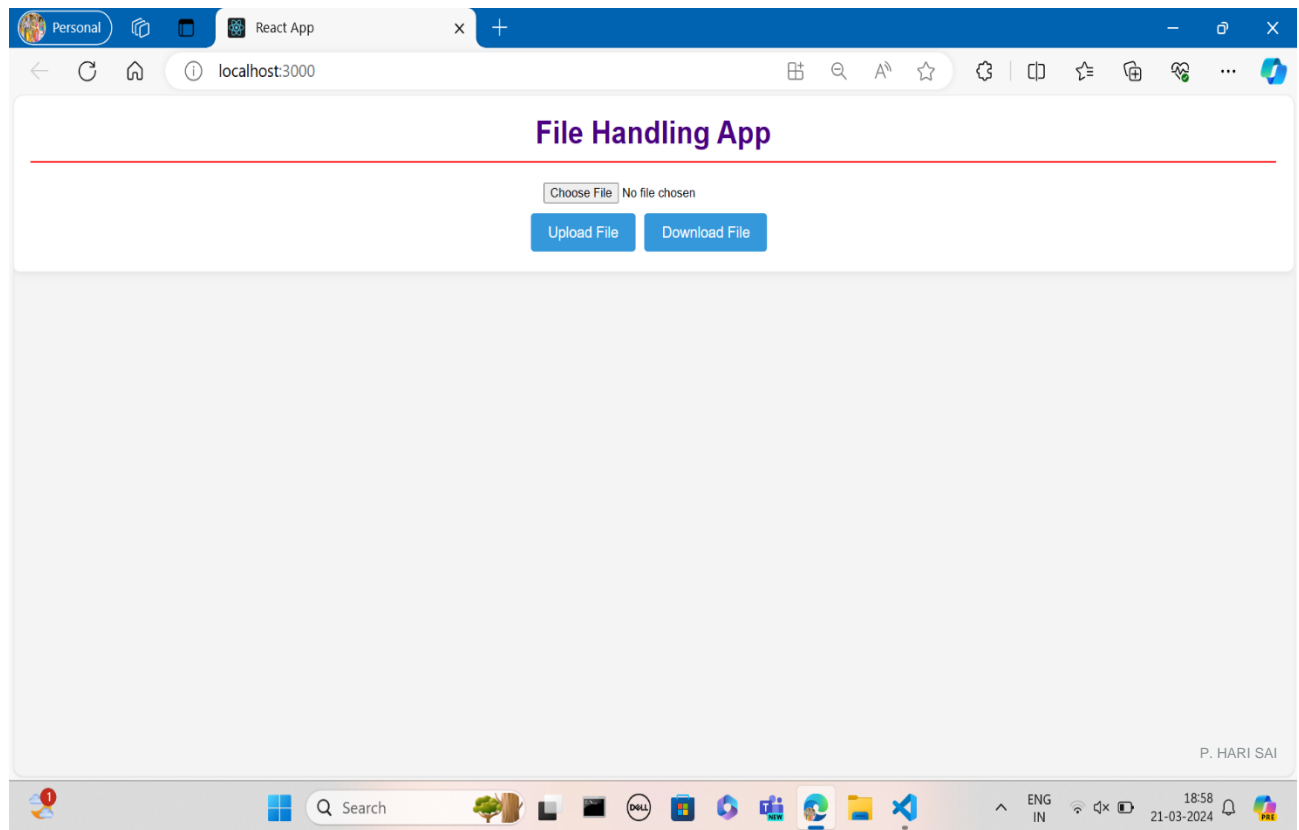
message
-----
User deleted

PS D:\React.js>
```

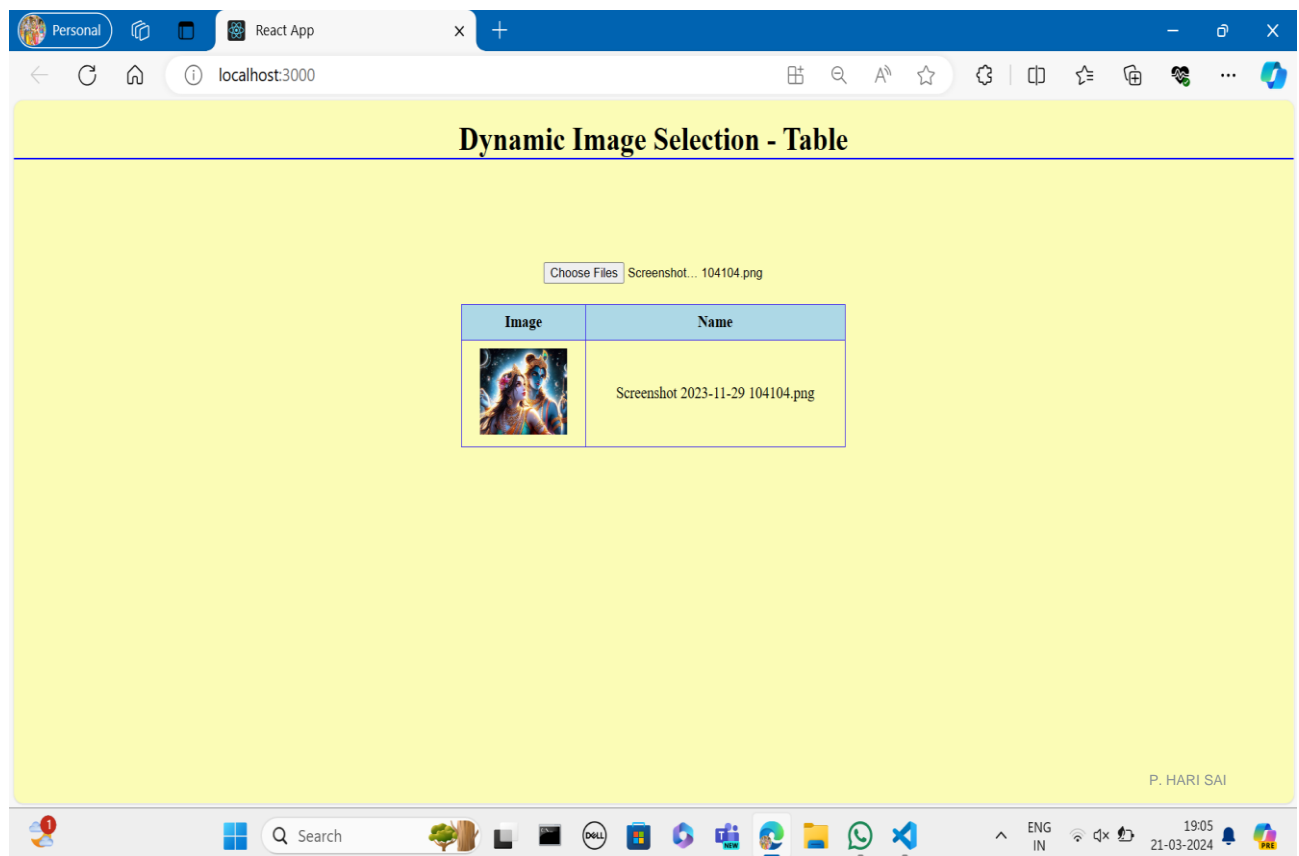
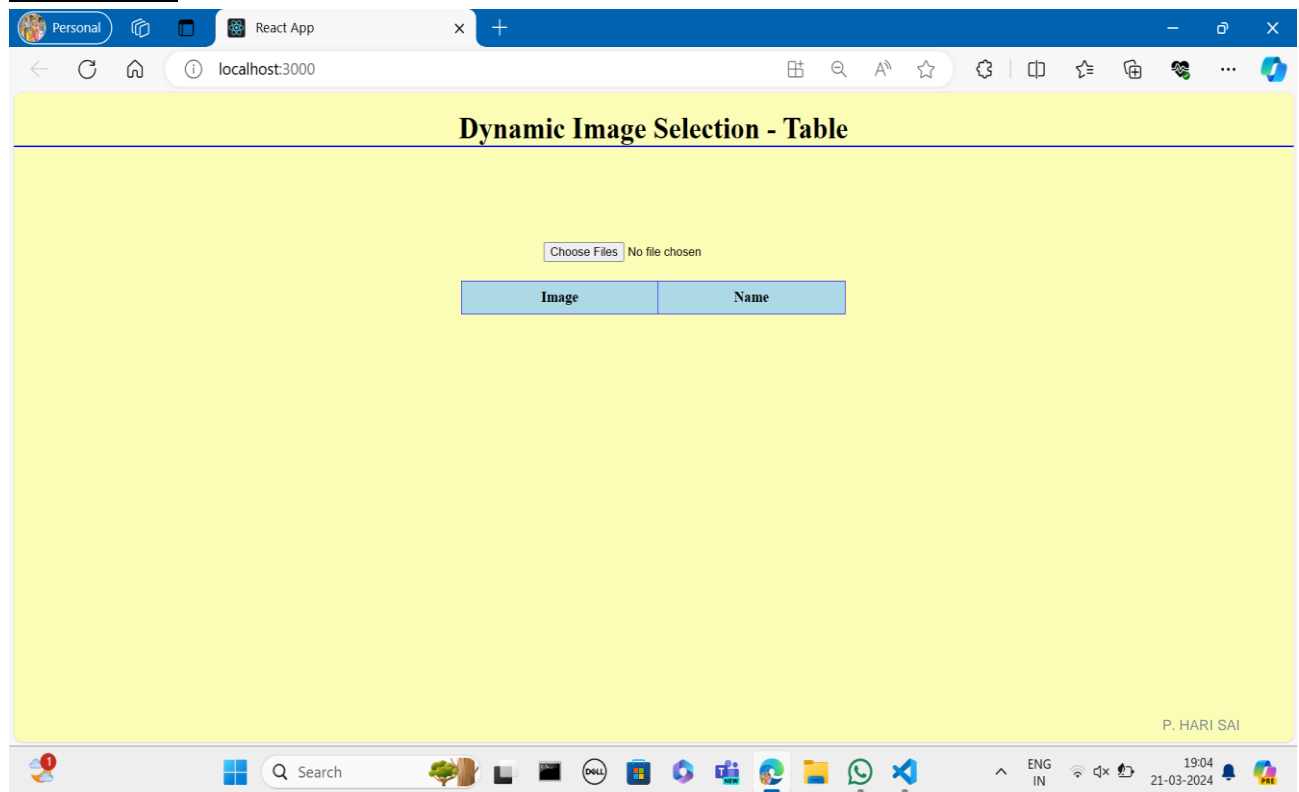


OUTPUT:









OUTPUT:



Personal React App localhost:3000

Dynamic Image Selection - Table

Choose Files Screenshot... 145056.png

	
	Screenshot 2024-01-27 111502.png
	Screenshot 2023-08-30 144311.png
	Screenshot 2024-02-13 145056.png

P. HARI SAI

19:06 21-03-2024