

→ React, NodeJS and MySQL Integration

Page No.:

Date:

① importing necessary libraries

```
const express = require('express');  
const mysql = require('mysql');  
const cors = require('cors');  
const bcrypt = require('bcrypt');
```

→ ①

② initializing the app & handling middle ware

```
const app = express();  
app.use(cors());  
app.use(express.json());
```

→ ②

③ creating a connection

```
const db = mysql.createConnection({  
  host: "localhost",  
  user: "root",  
  password: "ashutosh",  
  port: 3306,  
  database: " "});
```

```
db.connect((err) => {  
  if (err) {  
    console.log("error in database", err);  
    return;  
  }  
  console.log("connected to database");  
});
```

④ app.get('/', (req, res) => { res.json("hello"); });

⑤ app.listen(8800, () => { console.log("connected to database"); backend });

Shot on OnePlus

Majri, Nagpur Division | 2024.07.11 23:23



* Suppose we have a table for student who are registering & logging.

① Registering

app.post("/register", async (req, res) => {

try {

const hashedPassword = await bcrypt.hash(req.body.password, 10);

const q = "insert into userprofile (fullname, email, telephone, Aadhaar, password) values (?)";

const values = [req.body.fullname, req.body.email, req.body.telephone, req.body.Aadhaar, hashedPassword,];

db.query(q, [values], (err, data) => {

if (err) return res.status(500).json({ error: "Ise" });

res.json({ success: true });

});

} catch (error) {

console.error("Error during registration", error);

res.status(500).json({ error: "Internal server error" });

}

});

② logging

app.post("/login", async (req, res) => {

const { email, password } = req.body;

if (!email || !password) {

return res.status(400).json({ error: "Please provide email and password" });

const user = await db.query("select * from userprofile where email = ? and password = ?");

if (user.length === 0) {

return res.status(401).json({ error: "Invalid email or password" });

const user = user[0];

return res.json({ success: true, user });

});

③

if error give 500 response, otherwise response success true

Shot on OnePlus
Maui, Nagpur Division | 2024.07.11 23:23



② login

①—

```
app.post("/login", async (req, res) => {
  const { Aadhar, Password } = req.body;
  const q = "select * from userprofile where adhar = ?";
  db.query(q, [Aadhar], async (err, data) => {
    if (err) return res.status(500).json({ error: " " });
    if (data.length === 0) {
      return res.status(401).json({ error: "Invalid credentials" });
    }
  });
}
```

condition 1

②—

```
const user = data[0];
const match = await bcrypt.compare(Password, user.password);

if (!match) {
  return res.status(401).json({ error: "IsC" });
}

res.json({ success: true, user: user });
```

condition 2

③ forgetting all users

```
app.get("/userprofile", (req, res) => {
```

```
  const q = "select * from 'userprofile'";
```

```
  db.query(q, (err, data) => {
```

```
    if (err) return res.json(err);
```

```
    return res.json(data);
```

```
  });
}
```



★ React, Node.js & MySQL API's

Page No.:

Date:

for adding a user detail it is like same of the registration flow working

① books example

① Get all books

```
app.get("/books", (req, res) => {  
  const q = "select * from books"  
  db.query(q, (err, data) => {  
    if (err) return res.json(err)  
    return res.json(data)  
  })  
})
```

② inserting a book

```
app.post("/books", (req, res) => {  
  const q = "insert into books ('title', 'desc',  
    'price', 'cover') values (?)"  
  const values = [  
    req.body.title,  
    req.body.desc,  
    req.body.price,  
    req.body.cover  
  ]  
  db.query(q, [values], (err, data) => {  
    if (err) return res.json(err);  
    return res.json("book has been  
    created");  
  })  
})
```

③ deleting

```
app.delete("/books/:id", (req, res) => {  
  const bookId = req.params.id;  
  const q = "delete from books where id = ?"  
  db.query(q, [bookId], (err, data) => {  
    if (err) return res.send(err);  
    return res.json(data);  
  })  
})
```

④ app.put

```
app.put("/books/:id", (req, res) => {  
  const bookId = req.params.id;  
  const q = "update books set 'title' = ?,  
    'desc' = ?, 'price' = ?, 'cover' = ?  
    where id = ?"  
  const values = [  
    req.body.title, req.body.desc,  
    req.body.price, req.body.cover  
  ]  
  db.query(q, [...values, bookId],  
    (err, data) => {  
    if (err) return res.send(err);  
    return res.json(data);  
  })  
})
```

