

1. Create a new folder
2. Open the folder
3. Type **cmd** in Address bar
4. Type **npx create-react-app <projectname>**

**Eg: npx create-react-app studentapp**

5. Cd projectname (Open the Project)
6. Type code . (To open VS Code)
7. To Run your application with Default values –

Terminal -> New Terminal -> **npm start**

8. Create a new folder named **components** in **/src**
9. Create a new File Home.jsx (First letter must be Uppercase)
10. Type rafce in Home.jsx

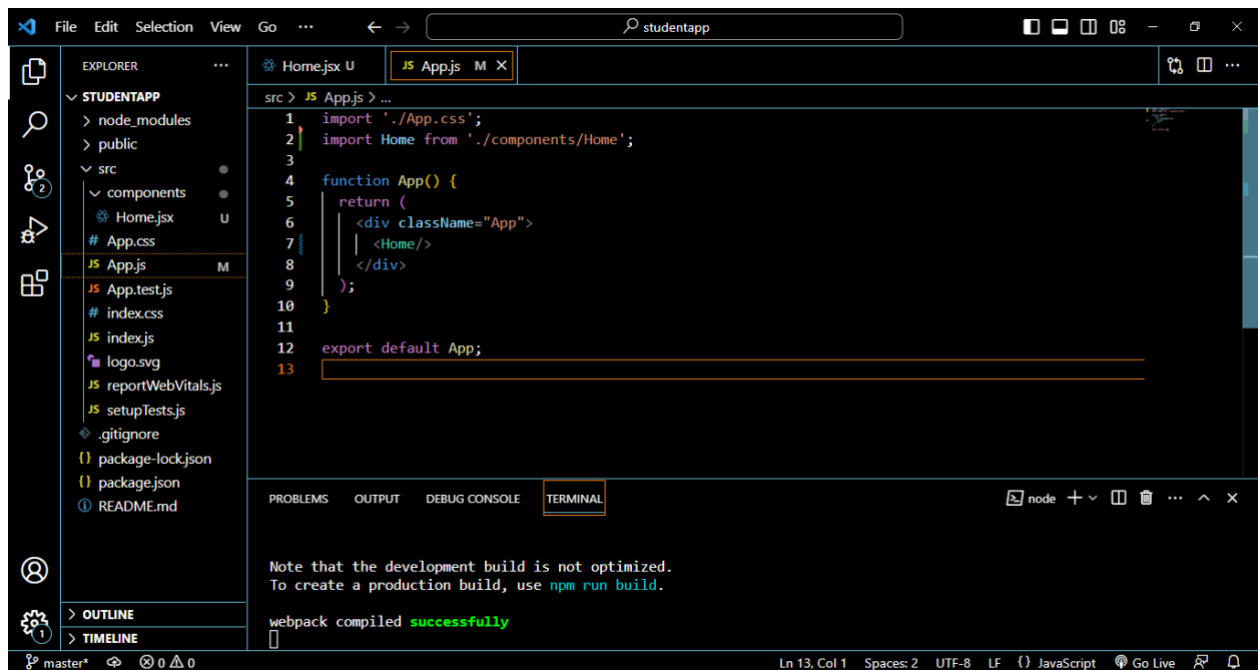


11. Type html code in Home.jsx in <div> tag

**<h1>Welcome to Student Application</h1>**

12. Open App.js (remove <header> tag)

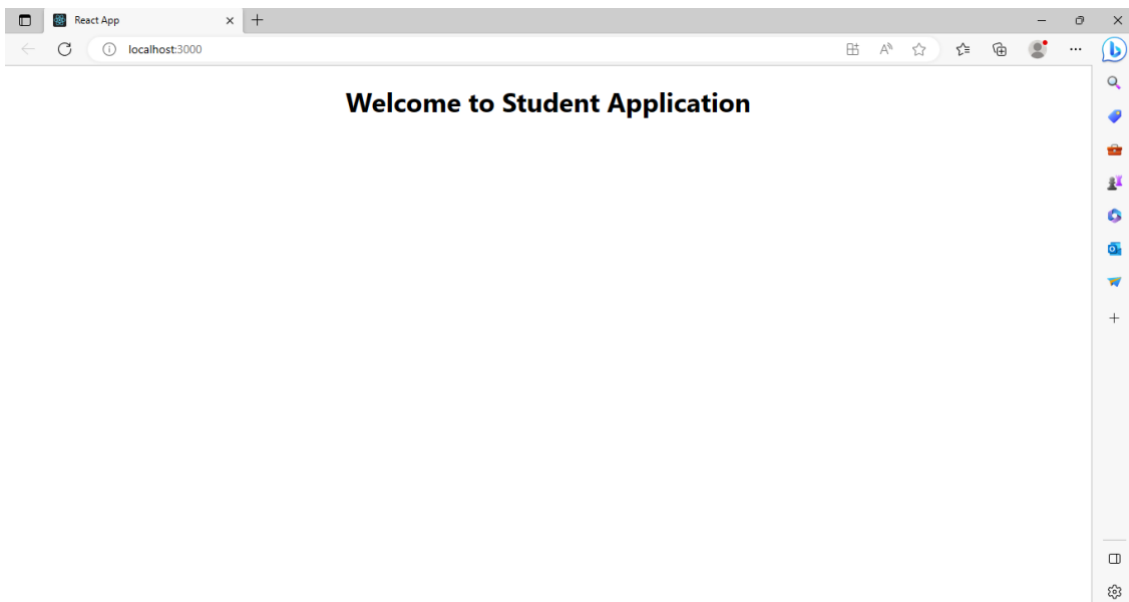
13. Type <Home/>



```
1 import './App.css';
2 import Home from './components/Home';
3
4 function App() {
5   return (
6     <div className="App">
7       <Home/>
8     </div>
9   );
10 }
11
12 export default App;
```

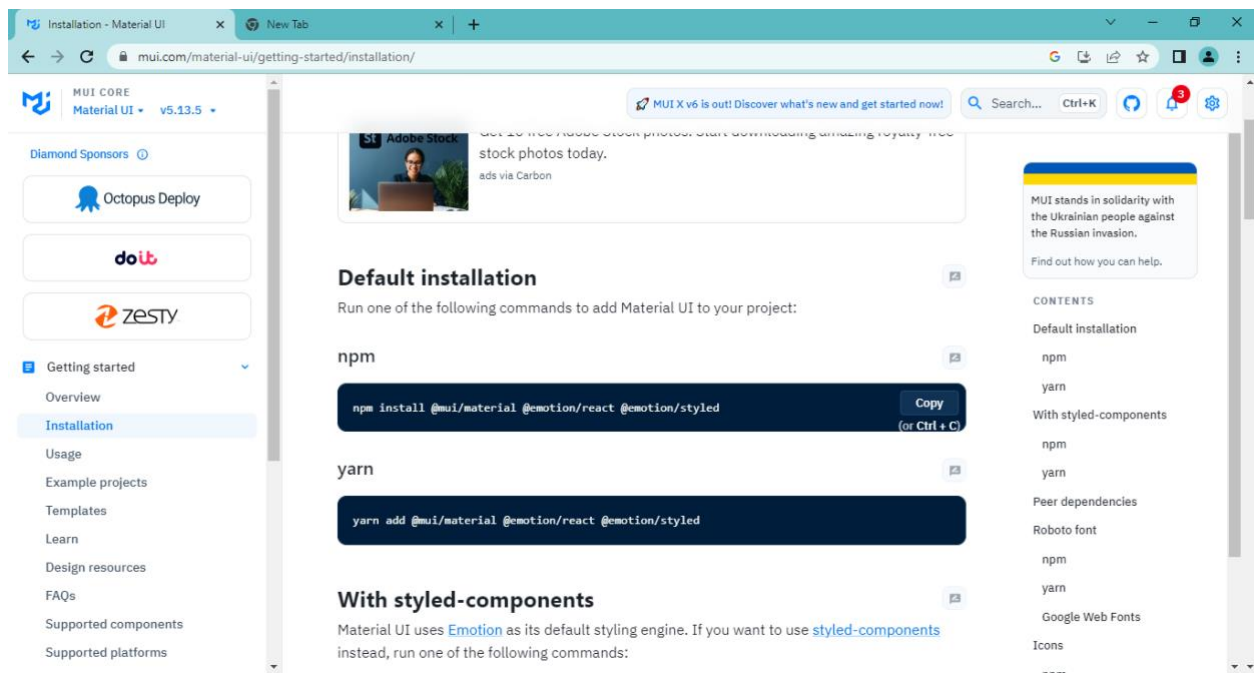
webpack compiled **successfully**

14. Open Browser and refresh it



15. Install Material UI

Search material ui in Google



Copy

**npm install @mui/material @emotion/react @emotion/styled**

**Type the above in Terminal**

16. Design the Home Page

```
import AppBar from '@mui/material/AppBar';
import Box from '@mui/material/Box';
import Toolbar from '@mui/material/Toolbar';
import Typography from '@mui/material/Typography';
import Button from '@mui/material/Button';
import IconButton from '@mui/material/IconButton';
import MenuIcon from '@mui/icons-material/Menu';

const Home = () => {

  return (
    <div>
      <Box sx={{ flexGrow: 1 }}>
        <AppBar position="static">
          <Toolbar>
            <IconButton
              size="large"
              edge="start"
              color="inherit"
```

```

        aria-label="menu"
        sx={{ mr: 2 }}
    >
        <MenuIcon />
    </IconButton>
    <Typography variant="h6" component="div" sx={{ flexGrow: 1 }}>
        Student Application
    </Typography>
    <Button color="inherit">Register</Button>
    <Button color="inherit">SIGN UP</Button>
</Toolbar>
</AppBar>
</Box>
</div>
)
}

export default Home

```

App.js (call Home.jsx)

```

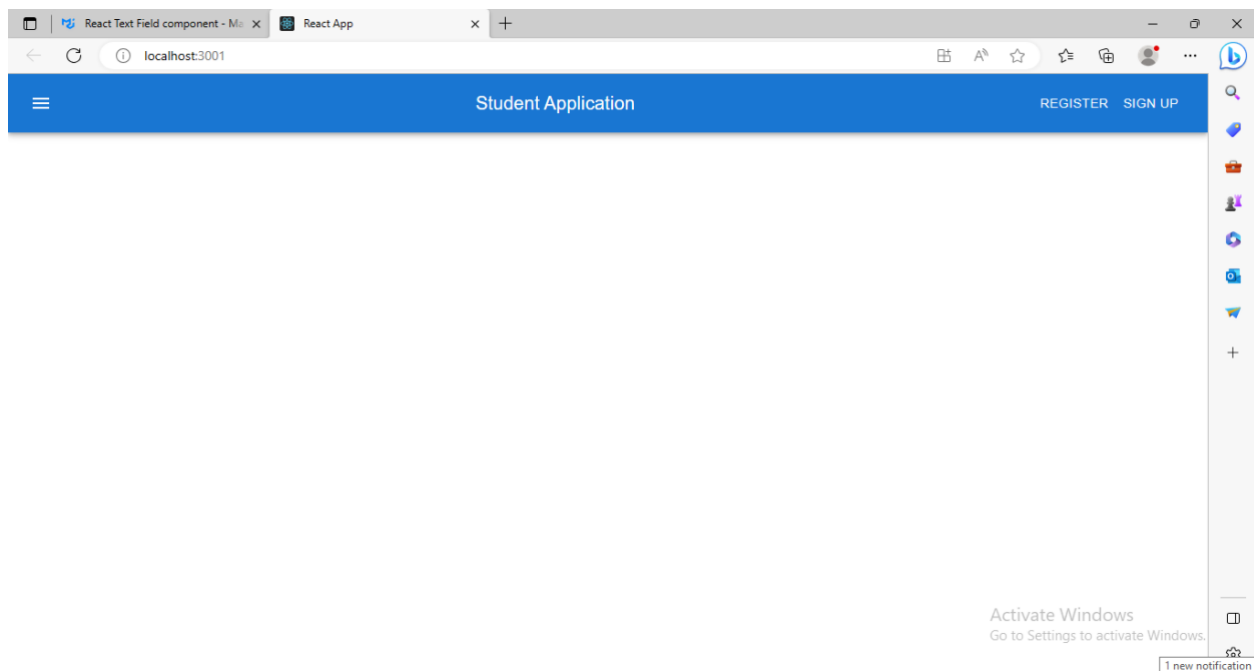
import './App.css';
import Home from './components/Home';

function App() {
    return (
        <div className='App'>
            <Home/>
        </div>
    );
}

export default App;

```

17. Run in terminal (npm start)



## 18. Design Student Page

```
import { Button, FormControl, FormControlLabel, FormLabel, Radio,
RadioGroup, TextField, Typography } from '@mui/material'
import React from 'react'

const Student = () => {
  return (
    <div>
      <Typography variant='h5'>Add Students</Typography><br></br>
      <TextField id="outlined-basic" label="ADMISSION NO"
variant="outlined" />
      <br></br>
      <br></br>
      <TextField id="outlined-basic" label="STUDENT NAME"
variant="outlined" />
      <br></br>
      <br></br>

      <FormControl>
        <FormLabel id="demo-radio-buttons-group-
label">Gender</FormLabel>
        <RadioGroup
          aria-labelledby="demo-radio-buttons-group-label"
          defaultValue="female"
          name="radio-buttons-group"
        >
          <Radio type="radio" value="female" /> Female
          <Radio type="radio" value="male" /> Male
        </RadioGroup>
      </FormControl>
    </div>
  )
}
```

```

        <FormControlLabel value="female" control={<Radio />}
label="Female" />
        <FormControlLabel value="male" control={<Radio />} label="Male"
/>
        <FormControlLabel value="other" control={<Radio />}
label="Other" />
    </RadioGroup>
</FormControl>
    <br></br>
    <br></br>
    <TextField id="outlined-basic" label="DOB" variant="outlined"
/>
    <br></br>
    <br></br>
    <Button variant='contained' color='success' >SUBMIT</Button>
</div>

)
}

export default Student

```

19. Terminal (install react router dom)

npm i react-router-dom

20. Open App.js and implement Router (To Access localhost:3000/Student)

```

import { BrowserRouter, Route, Routes } from 'react-router-dom';
import './App.css';
import Student from './components/Student';
import Home from './components/Home';

function App() {
  return (
    <div className='App'>
      <BrowserRouter>
        <Home/>
        <Routes>
          <Route path={'/Student'} element={<Student/>}></Route>
        </Routes>
      </BrowserRouter>
    </div>
  );
}

```

```
);
}

export default App;
```

21. Link to Register Button for Student.jsx

```
<Button color="inherit"><Link to='/Student'>Register</Link></Button>
```

## Mongo DB

22. Search MongoDB Atlas in google – Try Free – Sign in through Google

23. When you open MongoDB - Click on Browse Collections

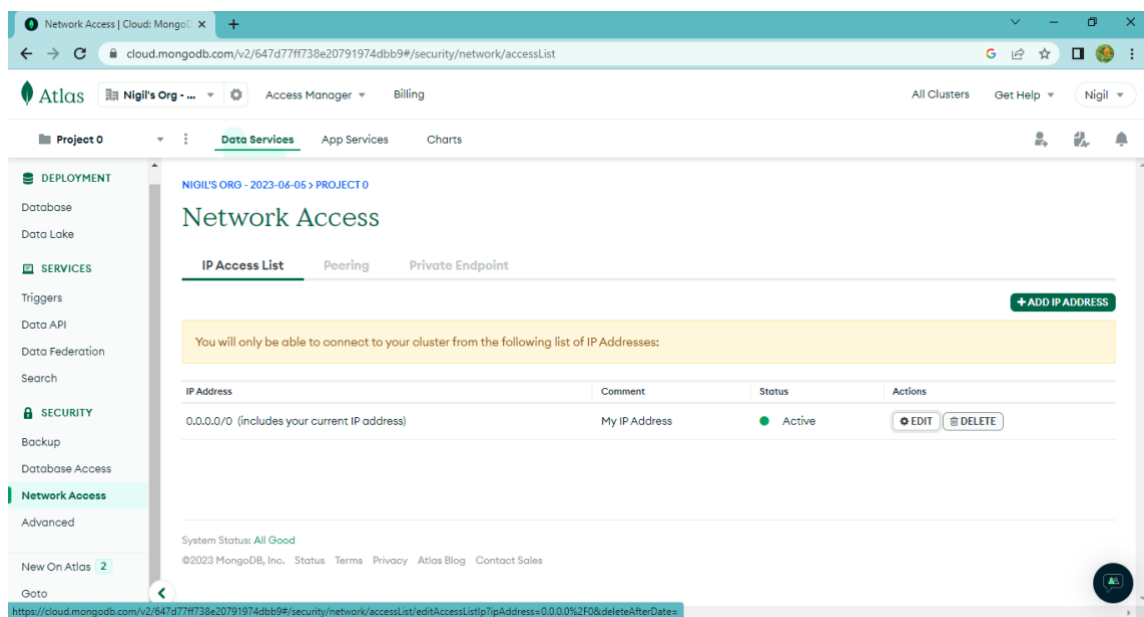
24. To create a new Database – Click on **Create Database**

A window will appear.

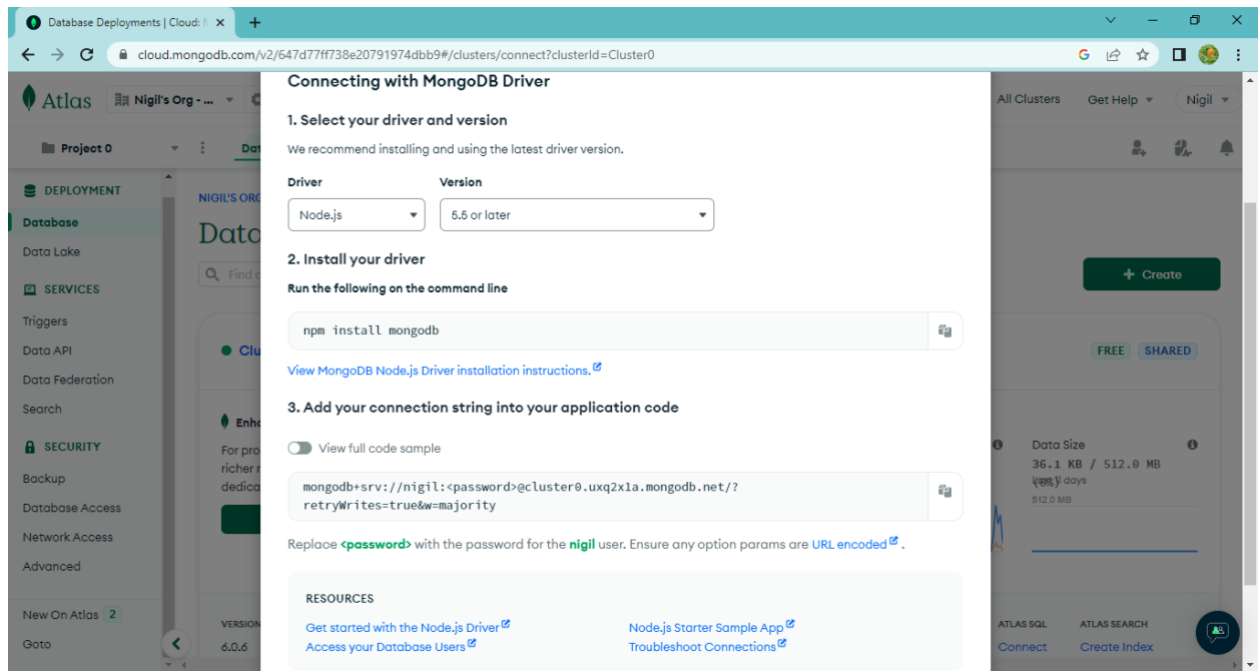
Type Database name and Collection name (table) and click on create button

25. In MongoDB, Click on Back up in SECURITY Section.

26. Click on IP Address – Click on Edit button - Change the IP address as **0.0.0.0/0**



27. Click on Database in Deployment section - Click on Connect – Drivers –



## 28. Copy the Connection String

mongodb+srv://nigil:<password>@cluster0.uxq2x1a.mongodb.net/?retryWrites=true&w=majority

# Backend

29. Create a folder named backend

30. Open it in VS Code

31. Install **npm init** in Terminal

32. If you want, given the author otherwise press enter key. Answer the question "Is it OK?" yes.

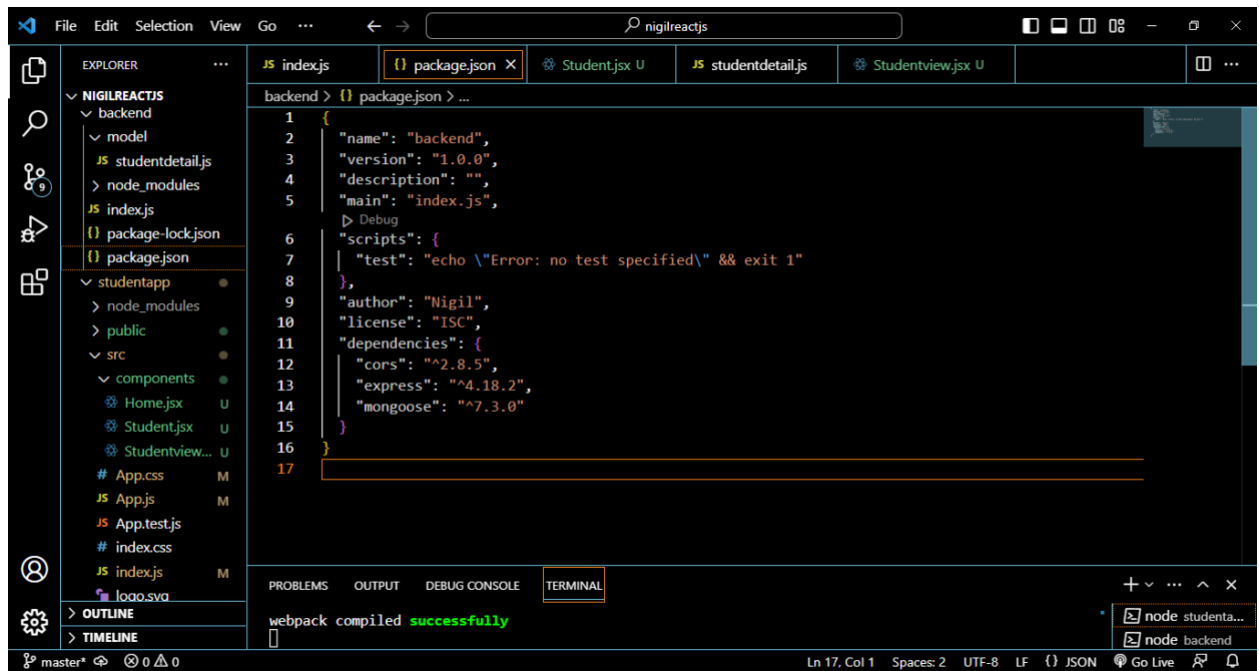
Package.json is created

33. Install **npm i mongoose** in Terminal

34. Install **npm i express** in Terminal

35. Install **npm i cors** in Terminal





Check the dependencies

36. Create a new file index.js and type the code

```
// import express
const express=require("express")
//Initailizing
const app=new express();

//Assign Port
app.listen(3005,(request,response)=>{
  console.log("Port is running in 3005")
})

//Api Creation
app.get('/',(request,response)=>{
  response.send("hai")
})
```

37. Run in Terminal

node index.js

check in Browser (localhost:3005). Hai appers in the screen

38. Create a new Folder named **model**

39. Create a new file named **studentdetails.js** in model

```
const mongoose=require("mongoose")
mongoose.connect("mongodb+srv://nigil:nigil@cluster0.uxq2x1a.mongodb.net/?retryWrites=true&w=majority")
.then(()=>{console.log("DB connected")})
.catch(err=>console.log(err));

let sc=mongoose.Schema;
const studschema = new sc(
  {
    Admno:Number,
    Sname:String,
    Gender:String,
    Dob:String
  }
);

var studmodel =mongoose.model("stud",studschema)
module.exports=studmodel;
```

40. Modify index.js as

```
// import express
const express=require("express")
const cors=require("cors")

//Initailizing
const app=new express();

const studmodel = require('./model/studentdetail')

app.use(express.urlencoded({extended:true}))
app.use(express.json());
app.use(cors());

//Api Creation
app.get('/',(request,response)=>{
  response.send("hai nigil")
})
//For Submit button
app.post('/new',(request,response)=>{
  console.log(request.body)
  new studmodel(request.body).save();
  response.send("Record Sucessfully Saved")
})
```

```
//Assign Port
app.listen(3005,(request,response)=>{
  console.log("Port is running in 3005")
})
```

#### 41. Update Student.jsx in studentapp

```
import { Button, TextField, Typography } from '@mui/material'
import axios from 'axios'
import React, { useState } from 'react'

const Student = () => {

  var [inputs,setInputs]
=useState({"Admno":"","Sname":"","Gender":"","Dob":""})

  const inputHandler = (e) => {
    const {name,value}=e.target
    setInputs((inputs) => ({...inputs,[name]:value}))
    console.log(inputs)
  }

  const addHandler =() =>{
    console.log("Clicked")

    console.log(inputs)
    axios.post("http://localhost:3005/new",inputs)
    .then((response) =>{
      alert("Record Saved")
    })
    .catch(err=>console.log(err))
  }

  return (
    <div>
      <Typography variant='h5'>Add
Students</Typography><br></br>
      <TextField id="outlined-basic" label="ADMISSION NO"
variant="outlined"
name="Admno" value={inputs.Admno}
onChange={inputHandler} />
      <br></br>
      <br></br>
    </div>
  )
}
```

```

        <TextField id="outlined-basic" label="STUDENT NAME"
variant="outlined"
        name="Sname" value={inputs.Sname}
onChange={inputHandler} />
        <br></br>
        <br></br>
        <TextField id="outlined-basic" label="Gender"
variant="outlined"
        name="Gender" value={inputs.Gender}
onChange={inputHandler} />
        <br></br>
        <br></br>

        <br></br>
        <br></br>
        <TextField id="outlined-basic" label="DOB"
variant="outlined" name="Dob" value={inputs.Dob}
onChange={inputHandler} />
        <br></br>
        <br></br>
        <Button variant='contained' color='success'
onClick={addHandler} >SUBMIT</Button>
    </div>

    )
}

export default Student

```

#### 42. Update App.js

```

<Route path='/Student' element={<Student method='post' />}/>

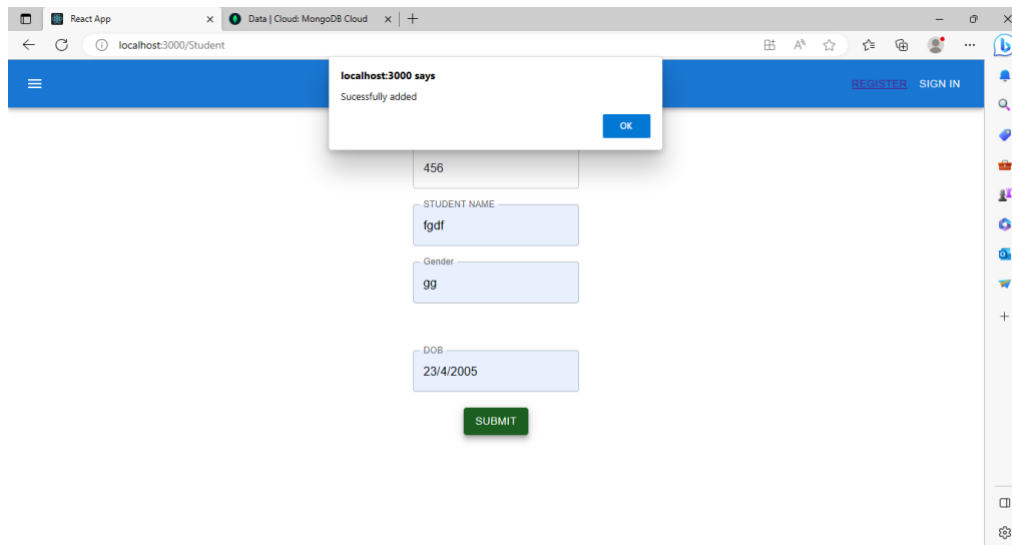
```

#### 43. Run in Terminal

studentapp - npm start

backend – nodemon index.js

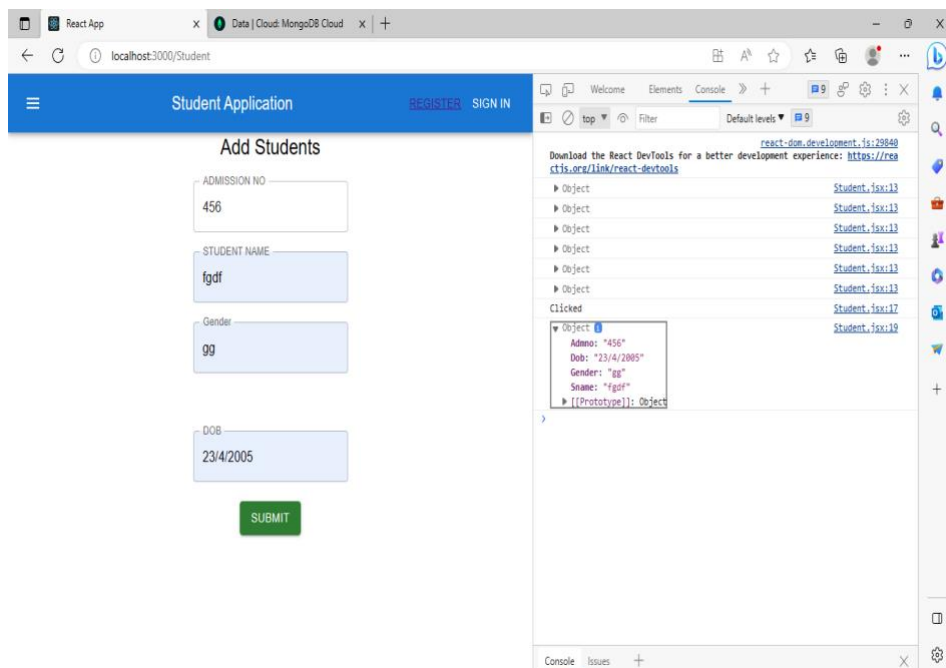
Output



## Inspect

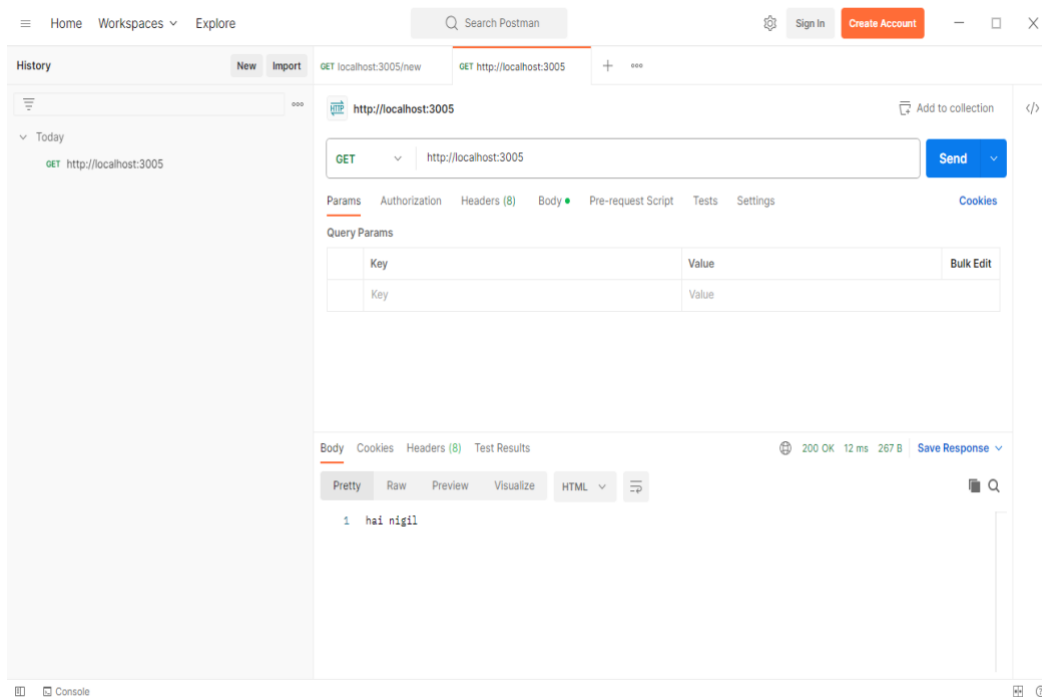
If you want to check the values in **console**

- Right click on Browser and take inspect
- Select Console



## POSTMAN

- Install Postman in your system
- Open Postman
- To Check GET – Select **Get** and type <http://localhost:3005> and click on **send** button



- To Check POST – for Submit

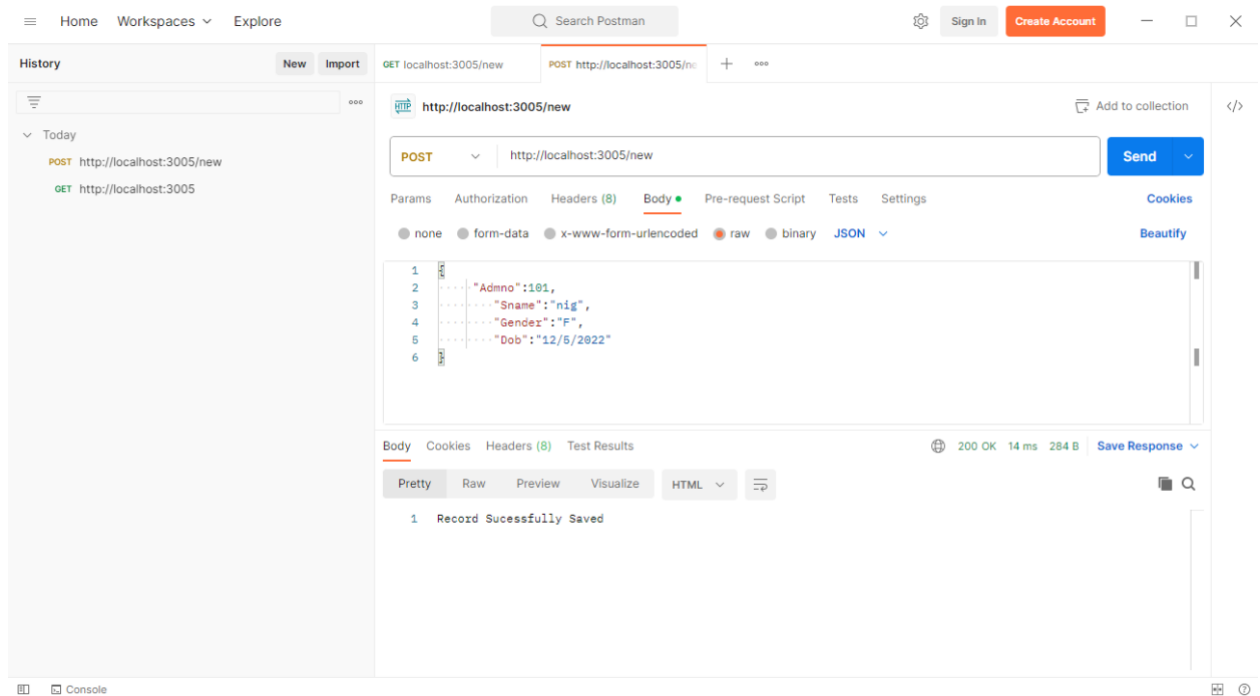
Select **POST** and type <http://localhost:3005/new>

Click on body – JSON

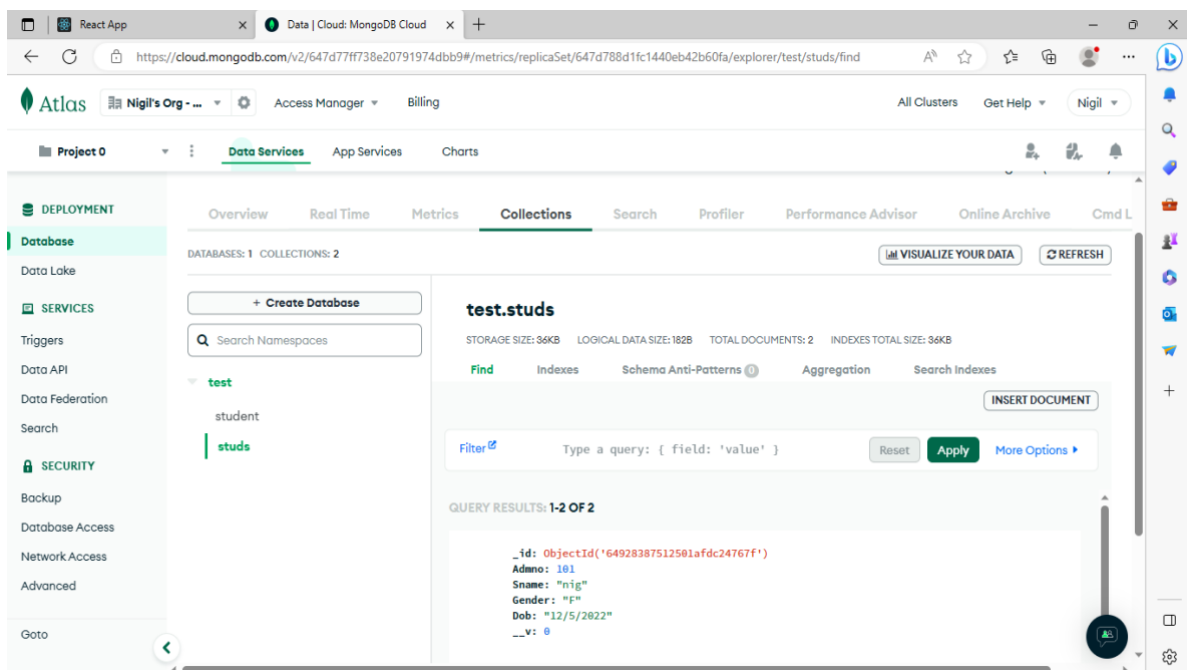
Type the values

```
{  
  "Admno":101,  
  "Sname":"nig",  
  "Gender":"F",  
  "Dob":"12/5/2022"  
}
```

and click on **send** button



- Check in MongoDB



## Student View

44. Create a new File Studentview.jsx in src/components

```
import React, { useEffect, useState } from 'react'
```

```

import { Typography } from '@mui/material';
import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TableRow from '@mui/material/TableRow';
import Paper from '@mui/material/Paper';
import axios from 'axios';

const Studentview = () => {
  var [students,setStudents] = useState([]);
  useEffect(()=>{
    axios.get("http://localhost:3005/view")
    .then(response =>{
      console.log(response.data)
      setStudents(response.data)    })
    .catch(err=>console.log(err))
  },[])

  return (
    <div>
      <Typography variant='h5'>Student
Details</Typography><br></br>
      <TableContainer component={Paper}>
        <Table sx={{ minWidth: 650 }} aria-label="simple table">
          <TableHead>
            <TableRow>
              <TableCell>Admission No</TableCell>
              <TableCell>Student Name</TableCell>
              <TableCell>Gender</TableCell>
              <TableCell>DOB</TableCell>

            </TableRow>
          </TableHead>
          <TableBody>
            {students.map((value,index)=>{
              return(
                <TableRow key={index}>
                  <TableCell>{value.Admno}</TableCell>
                  <TableCell>{value.Sname}</TableCell>
                  <TableCell>{value.Gender}</TableCell>
                  <TableCell>{value.Dob}</TableCell>
                </TableRow>
              )
            })}
          </TableBody>
        </Table>
      </TableContainer>
    </div>
  )
}

```



```

        </Table>
      </TableContainer>
    </div>
  )
}

export default Studentview

```

Update Home.jsx

```

<Routes>
  <Route path='/Student' element={<Student
method='post' />}/>
  <Route path='/StudentView' element={<Studentview
method='get' />}/>
</Routes>

```

45. Update index.js in backend

```

app.get('/view', async(request, response)=>{
  var data = await studmodel.find();
  response.send(data)
})

```

46. After Saving , just view the details ,modify Student.jsx

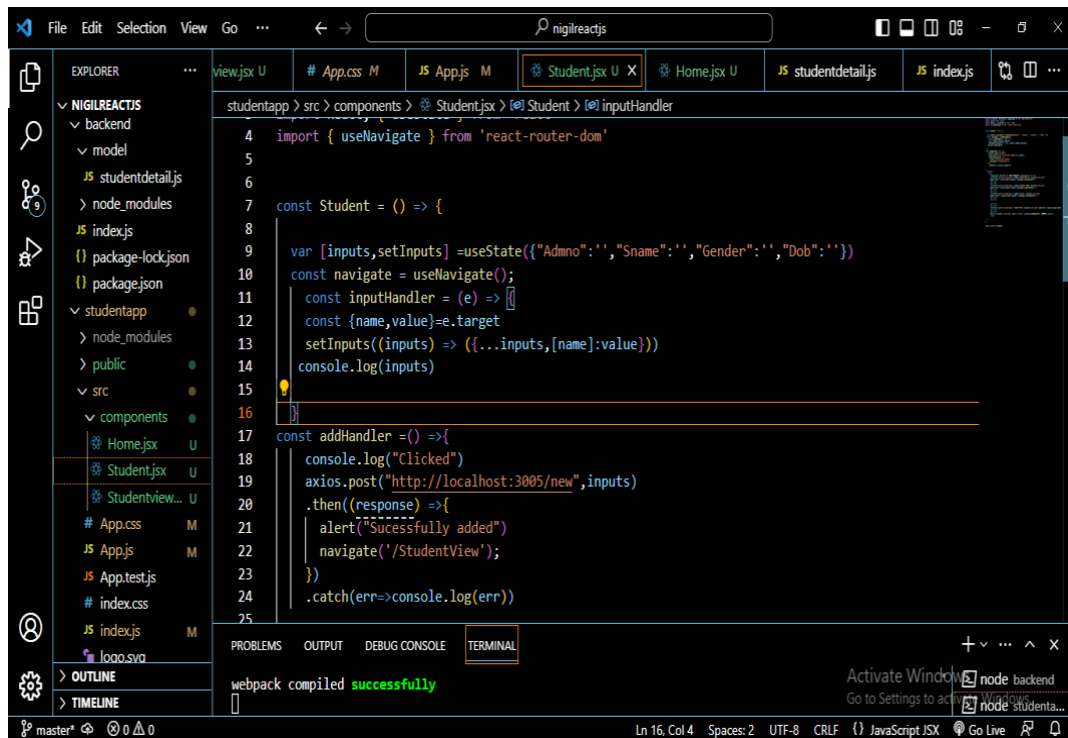
```

import { useNavigate } from 'react-router-dom'

//Student
const navigate = useNavigate();

//addHandler
navigate('/StudentView');

```



## Delete Operation

47. Create an icon for Delete and update in Studentview.jsx

```
import DeleteForeverIcon from '@mui/icons-material/DeleteForever';
import ModeEditIcon from '@mui/icons-material/ModeEdit';
```

```
<TableCell><ModeEditIcon color='success' /></TableCell>
<TableCell><DeleteForeverIcon color='error'
onClick={()=>deleteValues(value._id)} /></TableCell>
```

```
const deleteValues=(id)=>{
  console.log("deleted",id)
  axios.delete("http://localhost:3005/remove/"+id)
  .then((response)=>{
    alert("Deleted")
    //to reload window
    window.location.reload(false);
  })}
```

Update code in index.js

```
app.delete('/remove/:id', async(request, response)=>{
  let id = request.params.id
  await studmodel.findByIdAndDelete(id)
  response.send("Record deleted")
})
```

## Update Operation

48. Update Studentview.jsx

```
import Student from './Student';
var[selected,setSelected]=useState()
var[update,setUpdate]=useState(false)
```

```
const updateValues = (value) =>{
  console.log("updated:",value)
  setSelected(value);
  setUpdate(true);
}
```

```
var result=
  <TableContainer component={Paper}>
    <Table sx={{ minWidth: 650 }} aria-label="simple table">
      <TableHead>
        <TableRow>
          <TableCell>Admission No</TableCell>
          <TableCell>Student Name</TableCell>
          <TableCell>Gender</TableCell>
          <TableCell>DOB</TableCell>

        </TableRow>
      </TableHead>
      <TableBody>
        {students.map((value,index)=>{
          return(
            <TableRow key={index}>
              <TableCell>{value.Admno}</TableCell>
              <TableCell>{value.Sname}</TableCell>
              <TableCell>{value.Gender}</TableCell>
              <TableCell>{value.Dob}</TableCell>
              <TableCell><ModeEditIcon
color='success'onClick={()=>updateValues(value)}></TableCell>
```

```

                <TableCell><DeleteForeverIcon color='error'
onClick={()=>deleteValues(value._id)}/></TableCell>
            </TableRow>
        )
    }
}
</TableBody>

</Table>
</TableContainer>

```

```

if(update)
    result=<Student data={selected} method='put' />
    return (result)

```

#### 49. Update Student.jsx

```

const Student = (props) =>
{
    var [inputs,setInputs] =useState(props.data)
    console.log("method:",props.method)
    const navigate = useNavigate();
    const inputHandler = (e) =>
    {
        const {name,value}=e.target
        setInputs((inputs) => ({...inputs,[name]:value}))
        // console.log(inputs)
    }
}

```

props

```

const addHandler =() =>{
    console.log("Clicked")
    if(props.method === "post")
    {
        axios.post("http://localhost:3005/new",inputs)
        .then((response) =>{
            alert("Sucessfully added")
            navigate('/StudentView');
        })
        .catch(err=>console.log(err))
    }
}

```

```
if(props.method === 'put'){
  console.log("inside put")
  axios.put("http://localhost:3005/edit/"+ inputs._id,inputs)
  .then(response=>{
    console.log("post data"+response.data)
    alert("success")
    window.location.reload(false);
  })
  .catch(err=>console.log(err))
}

}
```

## 50. Update index.js

```
app.put('/edit/:id',async(request,response)=>
{
  let id = request.params.id;
  await studmodel.findByIdAndUpdate(id,request.body)
  response.send("Data updated")
})
```