

DAY-1

Form(html) :

Flipkart website-sankranti offers

latest fashion

Hurry up...!

from 15 to 20 dec

Buy your fashion clothes

1. **Casual Wear**
2. t-shirt
3. jeans
4. shorts

- First
- Second

favourite dish of Nashik:-MISAL PAV



Misal Pav is a popular and flavorful dish from Maharashtra, India, typically enjoyed as a breakfast or snack. It consists of two main components: Misal (a spicy curry) and Pav (bread rolls).

[link to Puranpoli wikipedia link](#)

Types of misal pav

1. Kolhapuri Misal
2. Pune Misal
3. Nashik Misal
4. Mumbai Misal

The Favourite Dish of Maharashtra is:-PURANPOLI



Puranpoli is a traditional Indian sweet flatbread, primarily associated with the festivals of Ganesh Chaturthi and Holi, particularly in the states of Maharashtra, Gujarat, and Karnataka. It is made by stuffing a dough with a mixture of jaggery and yellow lentils (usually toor dal or chana dal), flavored with cardamom and sometimes nutmeg. The dough is made from refined flour, and the filled dough is rolled into thin discs and cooked on a griddle, often with a little ghee for added flavor. The rich, sweet filling and soft, golden-brown exterior make puranpoli a beloved delicacy. The dish is typically enjoyed with milk or as part of festive meals, symbolizing prosperity and joy. It is a labor of love, as the preparation of both the dough and the filling requires attention and care, making it a cherished treat for family gatherings and celebrations.

Welcome to Yamifood Cafe

Fresh, Delicious meal to reach your optimum health and fitness

POCKET CAFE








[link to restaurant](#)

MAIN MENU

- Pizza
- Burger
- French Frise
- Cold Coffee

Home Delivery Availabale...!



SR.NO	NAME	PRICE	ORDER NO.	IMAGES
1.	PIZZA	100/-	005	
2.	Burger	80/-	3256	
3.	COLD COFFEE	100/-	3276	
4.	FRENCH FRIES	200/-	6756	
5.	SANDWICH	200/-	3456	

DAY-2

Task-1

HTML FORMS

Username :

Password :

Task 2:

REGISTRATION FORM

MR name:-

Age:-

Gender:

☐ Male

☐ Female

Date of birth:-

Religion:-

☐ Hindu

☐ Muslim

☐ Sikhim

☐ Other

CITY:-

XYZ College/School
Student Registration Form

Student Image:

No file chosen

(less than 5 MB)

Student Name:

Full Name

Father's Name:

Father's Full Name

Mother's Name:

Mother's Full Name

Gender:

☐ Male ☐ Female ☐ Other

Date of Birth:

mm/dd/yyyy



E-mail:

email@xyz.com

Level:

High School



Department:

Electrical Engineering

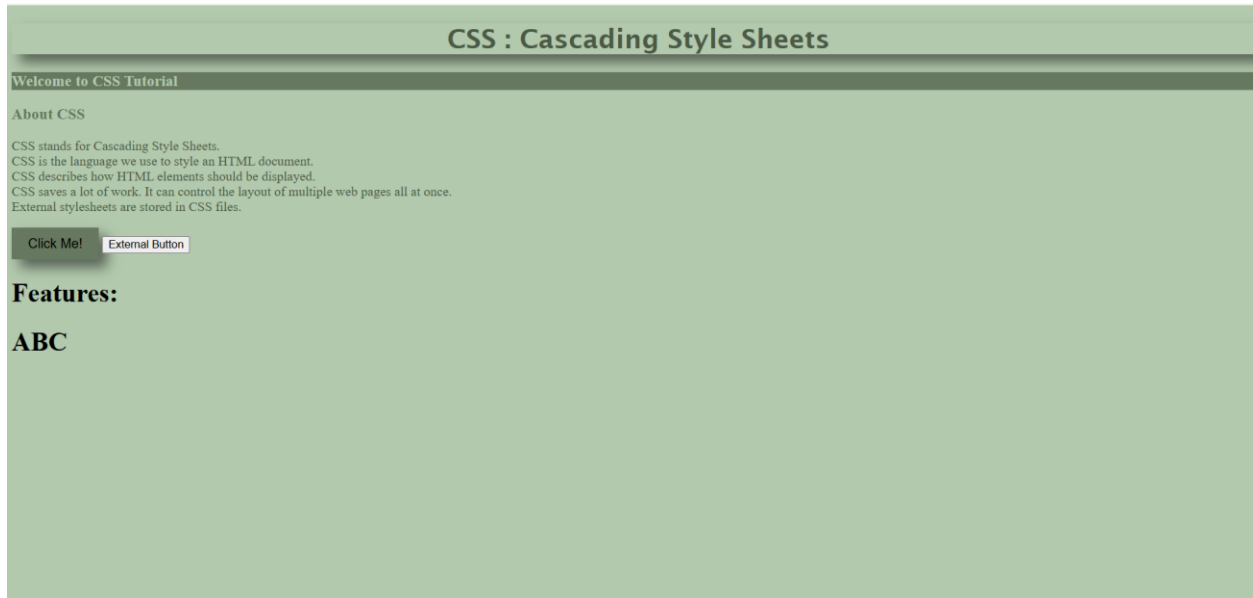


Tel/Mobile:

XXX XXX XXXX

DAY- 3

Task 1 :



Task 2:



DAY-4

Task-1

SHOP BY CATEGORY



Western Wear
50-80% OFF
[Shop Now](#)



Casual Wear
40-80% OFF
[Shop Now](#)



Kids Wear
40-80% OFF
[Shop Now](#)



Casual styles
40-80% OFF
[Shop Now](#)



Grooming
40-80% OFF
[Shop Now](#)



Womens Footwear
40-80% OFF
[Shop Now](#)



Casual Wear
40-80% OFF
[Shop Now](#)



Womens ActiveWear
40-80% OFF
[Shop Now](#)

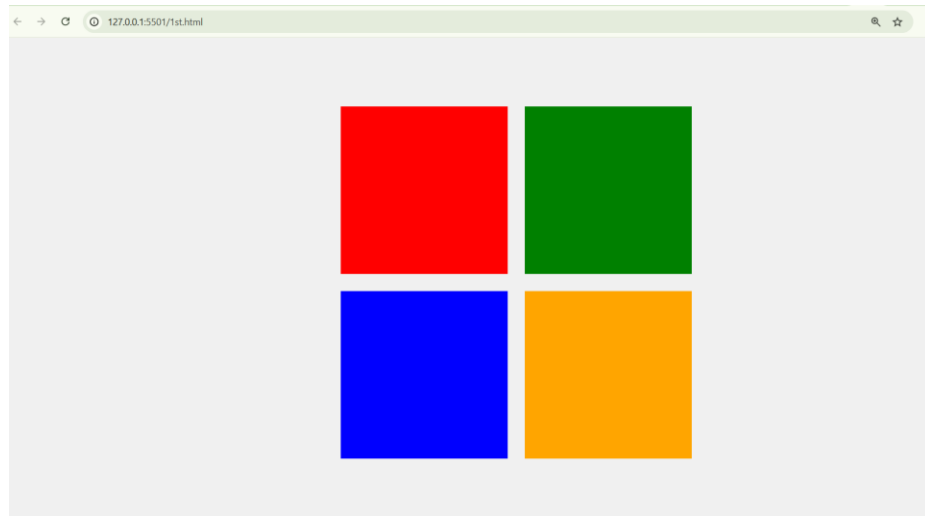


Mens Activewear
40-80% OFF
[Shop Now](#)



Sports Wear
40-80% OFF
[Shop Now](#)

Task-2



Task -3



DAY- 5

TASK-1 (JavaScript) :

```
C: > Users > DELL > Desktop > html > Next Gen > day 5 (JavaScript) > JS demo.js > ...
1  console.log("Hello World")
2
3  var a=10;
4  var b=20;
5  var c=a+b;
6  console.log(c)
7
8  console.log(4**4)
9  console.log(4*4)
10
11 console.log("-----");
12
13 let x=3
14 console.log("1 km =" + x * 0.621 + " miles")
15
16
17 let C = 2;
18 console.log("1 F = " + ((C * (9/5)) + 32) + " Fahrenheit")
19
20
21 let kg = 1;
22 console.log(kg + " kg = " + (kg * 2.20462) + " pounds");
```


OUTPUT PROBLEMS DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

```
Hello World
30
256
16
-----
1 km =1.863 miles
1 F = 35.6 Fahrenheit
1 kg = 2.20462 pounds
simple intrest =200
-----
under weight
-----
Rs 10 per unit charge=2000Rs
```

TASK-2 (BOOTSTRAP) :






NYKAA Fashion [Home](#) [Women](#) [Men](#) [All Brands](#) [Sort By](#) [Disabled](#)

NYKAA FASHION
REPUBLIC OF STYLE
SALE UP TO **80% OFF** **15TH - 21ST JAN** EXTRA **20% OFF***




SALE UP TO 80% OFF | 15th - 21st JAN

Sale is Live | Extra 20% off on your first order




TokyoZaraAllen SollyLibasLevis

GLOBAL STORE
Your one-stop shop for the latest trends from labels, around the world




Explore Global Store →

HOUSE OF NYKAA FASHION
Explore the latest collections from our most loved in house labels



Explore House of Nykaa →

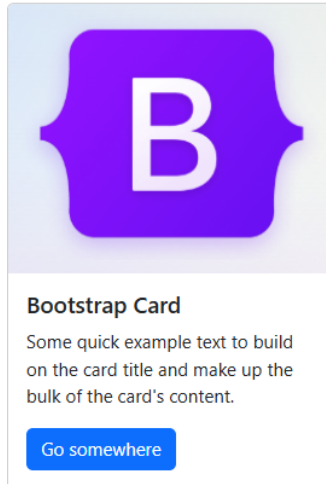
REVOLVE
Shop classic neutrals from 700+ celebrity approved brands



Explore Revolve →

Task -3 :

Bootstrap Practical



Example heading New



Weight (kg):

Height (m):

Your BMI is 30.22



DAY -6



DAY -7 (React)

Task -1 (calculator) :

The screenshot shows a development environment with VS Code on the left and a web browser on the right. In VS Code, the file explorer on the left shows a project structure with 'Project5' containing 'src' and 'components'. The 'components' folder is expanded, showing 'Component1.jsx', 'Component2.jsx', and 'Component3.jsx'. The main editor displays the code for 'Component1.jsx', which imports React and useState, and defines a 'Component1' function. The function sets up state for two numbers and a result, and defines a 'storenum1' function to handle input changes. The terminal shows the command 'npm run dev' and the output 'VITE v6.0.7 ready in 413 ms'. The web browser on the right shows the running application at 'localhost:5174'. It has a title 'Calculator' and a simple interface with two input fields (one containing '12', the other '2') and four buttons: 'ADD', 'SUB', 'MUL', and 'DIV'. Below the browser, the text 'Result: 24' is displayed.

```
1 import React, { useState } from 'react';
2
3
4 function Component1() {
5
6   const [num1, setnum1] = useState(0);
7   const [num2, setnum2] = useState(0);
8   const [res, setres] = useState(0);
9
10  const storenum1 = (e) => {
11    setnum1(parseFloat(e.target.value));
12    console.log(num1);
13  }
14
15  const storenum2 = (e) => {
```

C:\Users\DELL\Desktop\html\Next Gen\7day\Project5>npm run dev

> project5@0.0.0 dev
> vite

Port 5173 is in use, trying another one...

VITE v6.0.7 ready in 413 ms

→ Local: http://localhost:5174/
→ Network: use --host to expose
→ press h + enter to show help

Calculator

12
2
ADD SUB MUL DIV

Result: 24

Task -2(BMI Calculator) :

The screenshot shows a development environment with VS Code on the left and a web browser on the right. In VS Code, the file explorer on the left shows a project structure with 'Project5' containing 'src' and 'components'. The 'components' folder is expanded, showing 'Component1.jsx', 'Component2.jsx', and 'Component3.jsx'. The main editor displays the code for 'Component2.jsx', which defines a 'BMICalculator' function. The function has a 'getBMICategory' helper function that returns categories based on BMI values. The 'BMICalculator' function returns a JSX element with a centered heading 'BMI Calculator' and a label. The terminal shows the command 'npm run dev' and the output 'VITE v6.0.7 ready in 137 ms'. The web browser on the right shows the running application at 'localhost:5174'. It has a title 'Calculator' and a simple interface with two input fields (one containing '34', the other '12') and four buttons: 'ADD', 'SUB', 'MUL', and 'DIV'. Below the browser, the text 'Result: 408' is displayed. Below that, the text 'BMI Calculator' is displayed. Below that, the text 'Your BMI: 22.22' is displayed. Below that, the text 'Category: Normal weight' is displayed.

```
3 function BMICalculator() {
4   // Function to categorize BMI
5   const getBMICategory = () => {
6     if (bmi < 18.5) return 'Underweight';
7     if (bmi >= 18.5 && bmi < 24.9) return 'Normal weight';
8     if (bmi >= 25 && bmi < 29.9) return 'Overweight';
9     if (bmi >= 30) return 'Obesity';
10    return '';
11  };
12
13  return (
14    <div style={{ textAlign: 'center' }}>
15      <h1>BMI Calculator</h1>
16      <div>
17        <label>
```

C:\Users\DELL\Desktop\html\Next Gen\7day\Project5>npm run dev

> project5@0.0.0 dev
> vite

Port 5173 is in use, trying another one...

VITE v6.0.7 ready in 137 ms

→ Local: http://localhost:5174/
→ Network: use --host to expose
→ press h + enter to show help

Calculator

34
12
ADD SUB MUL DIV

Result: 408

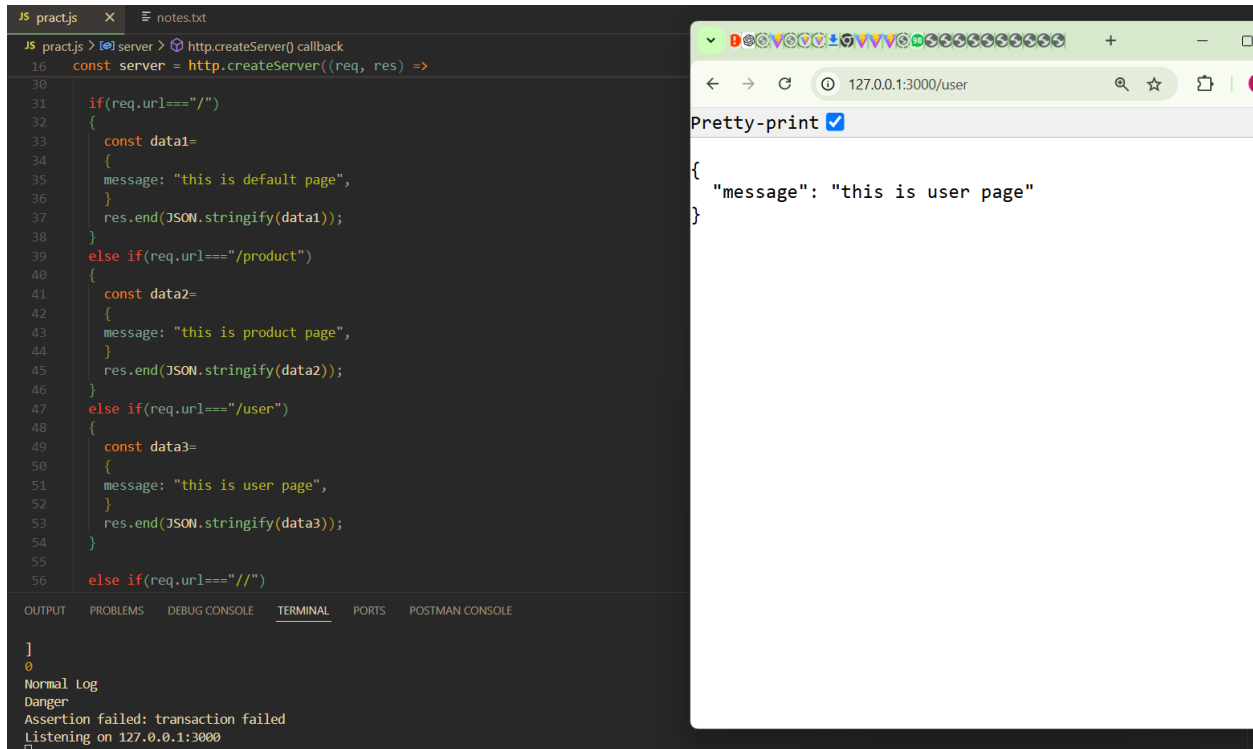
BMI Calculator

Weight (kg): 50
Height (m): 1.50
Calculate BMI

Your BMI: 22.22

Category: Normal weight

DAY -8 (NodeJS)

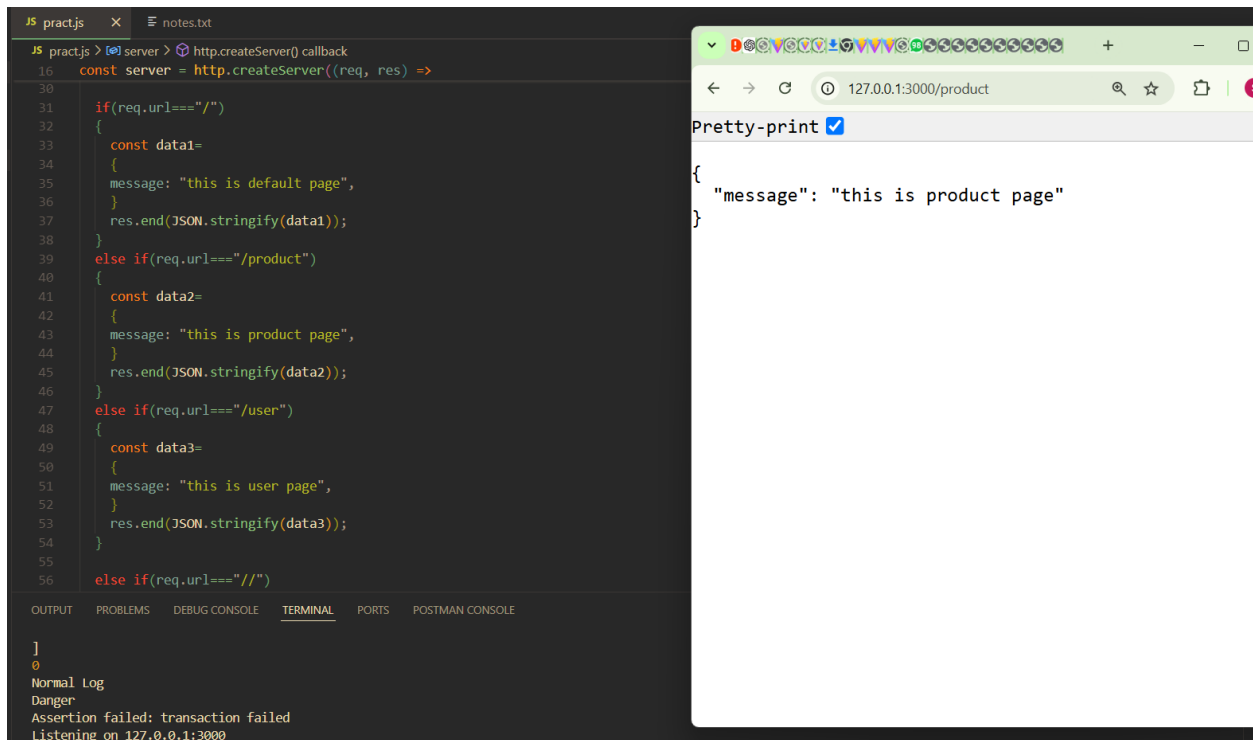


```
JS practjs X notes.txt
JS practjs > server > http.createServer() callback
16 const server = http.createServer((req, res) =>
30
31 if(req.url==="/")
32 {
33   const data1=
34   {
35     message: "this is default page",
36   }
37   res.end(JSON.stringify(data1));
38 }
39 else if(req.url==="/product")
40 {
41   const data2=
42   {
43     message: "this is product page",
44   }
45   res.end(JSON.stringify(data2));
46 }
47 else if(req.url==="/user")
48 {
49   const data3=
50   {
51     message: "this is user page",
52   }
53   res.end(JSON.stringify(data3));
54 }
55 else if(req.url==="/")
56
}
0
Normal Log
Danger
Assertion failed: transaction failed
Listening on 127.0.0.1:3000
```

127.0.0.1:3000/user

Pretty-print ☒

```
{
  "message": "this is user page"
}
```



```
JS practjs X notes.txt
JS practjs > server > http.createServer() callback
16 const server = http.createServer((req, res) =>
30
31 if(req.url==="/")
32 {
33   const data1=
34   {
35     message: "this is default page",
36   }
37   res.end(JSON.stringify(data1));
38 }
39 else if(req.url==="/product")
40 {
41   const data2=
42   {
43     message: "this is product page",
44   }
45   res.end(JSON.stringify(data2));
46 }
47 else if(req.url==="/user")
48 {
49   const data3=
50   {
51     message: "this is user page",
52   }
53   res.end(JSON.stringify(data3));
54 }
55 else if(req.url==="/")
56

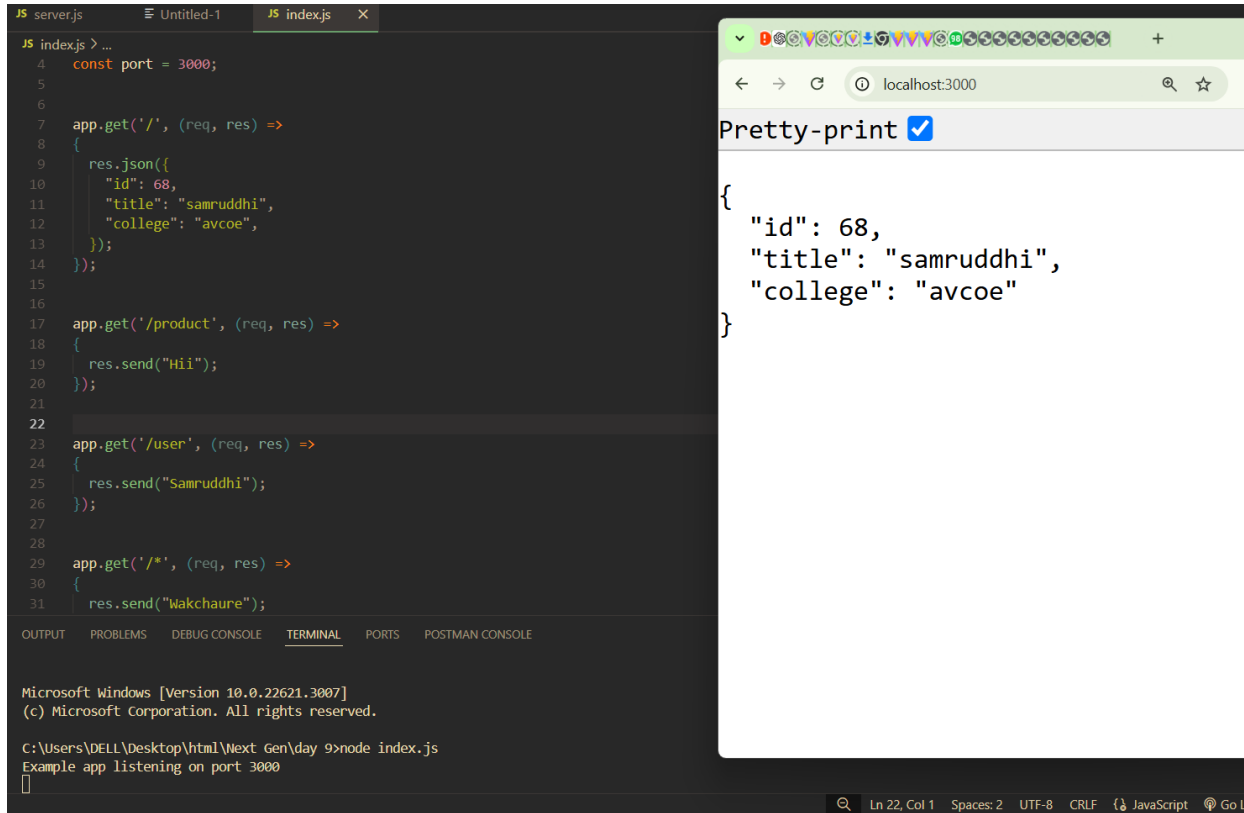
}
0
Normal Log
Danger
Assertion failed: transaction failed
Listening on 127.0.0.1:3000
```

127.0.0.1:3000/product

Pretty-print ☒

```
{
  "message": "this is product page"
}
```

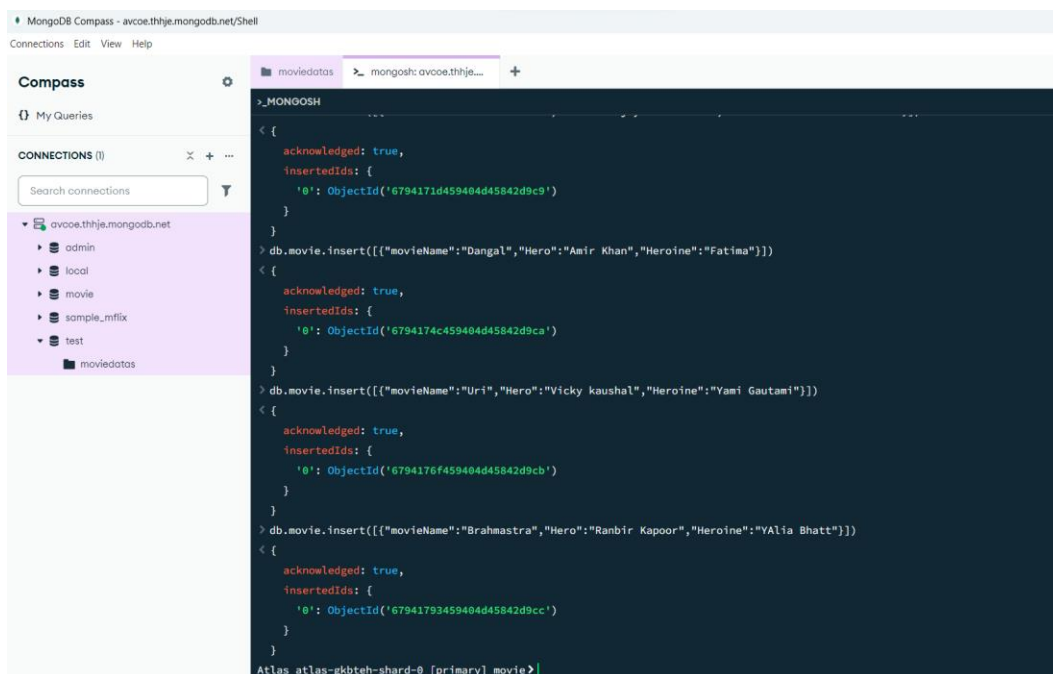
Day - 9



The screenshot shows a VS Code editor with a file named `index.js` containing a Node.js application. The application uses Express.js to create a web server listening on port 3000. It has three routes: a root route (`/`) that returns a JSON object, a `/product` route that sends "Hi!", and a `/user` route that sends "Samruddhi". There is also a catch-all route (`/*`) that sends "Wakchaure".

```
JS index.js > ...
4  const port = 3000;
5
6
7  app.get('/', (req, res) =>
8  {
9    res.json({
10     "id": 68,
11     "title": "samruddhi",
12     "college": "avcoe",
13   });
14 });
15
16
17 app.get('/product', (req, res) =>
18 {
19   res.send("Hi!");
20 });
21
22
23 app.get('/user', (req, res) =>
24 {
25   res.send("Samruddhi");
26 });
27
28
29 app.get('/*', (req, res) =>
30 {
31   res.send("Wakchaure");
32 });
```

The browser window shows the output of the root route (`localhost:3000/`), displaying the JSON object: `{ "id": 68, "title": "samruddhi", "college": "avcoe" }`. The browser's developer tools show the response in the "Pretty-print" tab.



The screenshot shows the MongoDB Compass interface. The left sidebar shows the "Connections" list with a connection to `avcoe.thlje.mongodb.net`. The main panel shows the `movie` collection in the `avcoe.thlje.mongodb.net` database. The collection contains five documents, each representing a movie with its name, hero, and heroine.

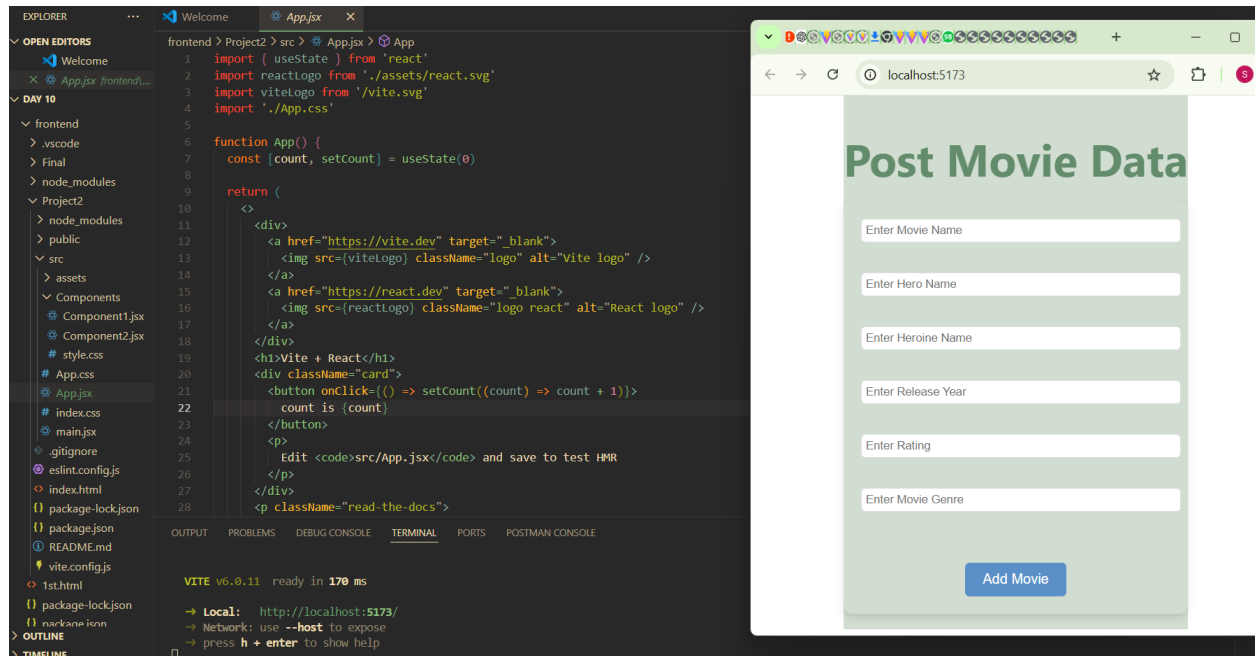
```
>_MONGODB_
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6794171d459404d45842d9c9')
  }
}
> db.movie.insert([{"movieName": "Dangal", "Hero": "Amir Khan", "Heroine": "Fatima"}])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6794174c459404d45842d9ca')
  }
}
> db.movie.insert([{"movieName": "Uri", "Hero": "Vicky kaushal", "Heroine": "Yami Gautam"}])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6794176f459404d45842d9cb')
  }
}
> db.movie.insert([{"movieName": "Brahmastra", "Hero": "Ranbir Kapoor", "Heroine": "YAlia Bhatt"}])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67941793459404d45842d9cc')
  }
}
Atlas atlas-gkbteh-shard-0 [primary] movie>
```

Pretty-print ☒

```
{
  "id": 1,
  "name": "John Doe",
  "email": "john.doe@example.com",
  "age": 30,
  "address": {
    "street": "123 Main St",
    "city": "Anytown",
    "state": "CA",
    "postalCode": "12345"
  }
},
{
  "id": 2,
  "name": "Jane Smith",
  "email": "jane.smith@example.com",
  "age": 25,
  "address": {
    "street": "456 Oak St",
    "city": "Othertown",
    "state": "TX",
    "postalCode": "67890"
  }
},
{
  "id": 3,
  "name": "Mike Johnson",
  "email": "mike.johnson@example.com",
  "age": 35,
  "address": {
    "street": "789 Pine St",
    "city": "Anothertown",
    "state": "NY",
    "postalCode": "54321"
  }
}
```


Day -10

Frontend :



get movie data

load movie data

Movie name	Hero	Heroin	year	Rating	genre
------------	------	--------	------	--------	-------

Backend:



```
backend > JS pract.js > ...
1  const express = require('express')
2  const cors = require('cors')
3
4  const app = express()
5  app.use(express.json());
6  app.use(cors)
7  const mongoose = require('mongoose');
8  const Moviemodel = require('./models/Moviemodel');
9  const port = 3000
10
11 app.get('/', (req, res) => {
12   res.send('Hi!')
13 })
14 app.post('/addmoviedata', async(req, res) => {
15   try{
16     const newdata=new Moviemodel(req.body);
17     await newdata.save();
18     console.log(req.body);
19     res.send("data saved")
20   }
21   catch(err){
22     console.log("data not saved")
23   }
24 })
25 app.get('/getmoviedata',async (req, res) => {
26   try{
27     const newdata = await Moviemodel.find({});
28     res.json(newdata);
```

OUTPUT PROBLEMS DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node pract.js`
Example app listening on port 3000
data connected successfully
```

Day -11

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 5 of 5

```
_id: ObjectId('6790ba41a629fbb66f7cecab')
name : "Dangal"
hero : "Aamir Khan"
heroine : "Sakshi Tanwar"
release_year : 2016
rating : 8.4
genre : "Biographical-Drama"
```

```
_id: ObjectId('6790ba41a629fbb66f7ceca9')
name : "Dilwale Dulhania Le Jayenge"
hero : "Shah Rukh Khan"
heroine : "Kajol"
release_year : 1995
rating : 8.1
genre : "Romance"
```

```
_id: ObjectId('6790ba41a629fbb66f7ceca8')
name : "3 Idiots"
hero : "Aamir Khan"
heroine : "Kareena Kapoor"
```

Post Movie Data

hmv
abjhb
kjhjvhg
202
7
thriller

Add Movie Data

localhost:5173 says
Data saved successfully

OK



KGF

Allu Arjun

kaniska

5



3 idiots

Amir khan

kareena kapoor

8.9

...

1

2

3

5

4

6

Know More

...

1

2

3

5

4

6

Know More