- **Eclipse IDE**
  - **Installation:**

https://eclipseide.org/

  - **Set – Up**

    File-Switch workspace

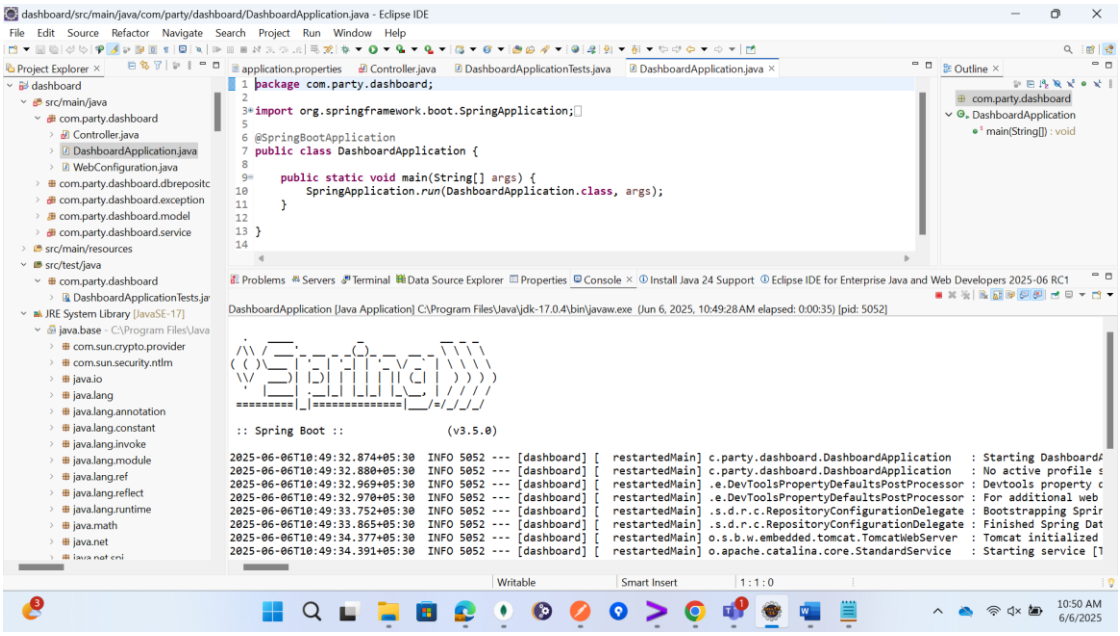
  - **Importing files**

    Import project-Existing projects into workspace

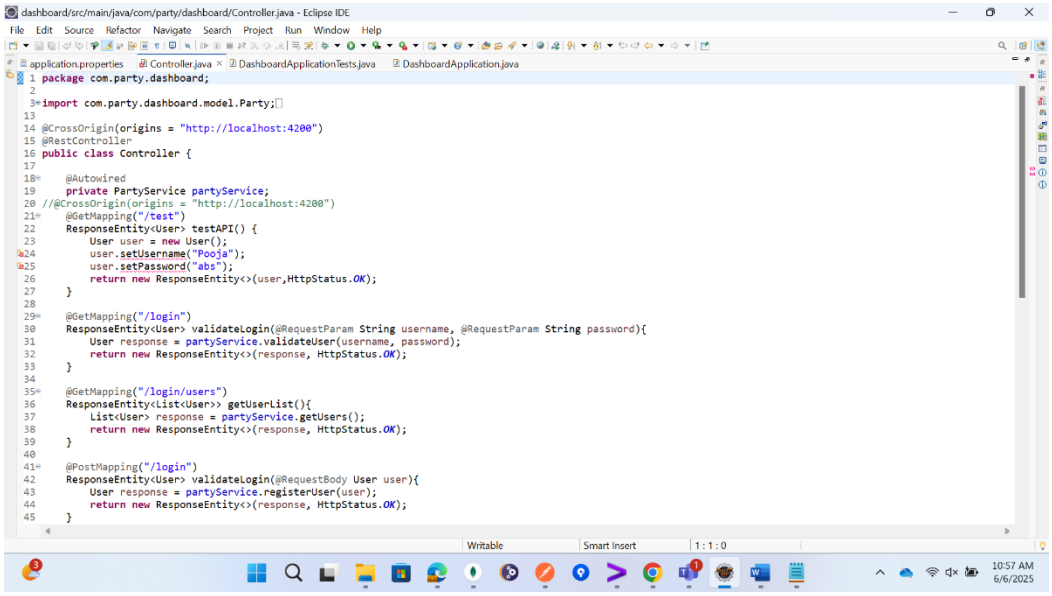
  - **Main class**

    Dashboardapplication.java

```
package com.party.dashboard;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class DashboardApplication {
        public static void main(String[] args) {
                SpringApplication.run(DashboardApplication.class, args);
        }
}
```



- Controller.java



- DashboardApplicationTests.java

```
package com.party.dashboard;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class DashboardApplicationTests {

	@Test
	void contextLoads() {
	}
}
```

- **Technologies Used in Your Spring Boot Project**

1.Backend Framework

-Spring Boot (v3.5.0)

https://start.spring.io/

2. Database

Mongodb

https://www.mongodb.com/try/download/community

3.Programming Language

Download Java JDK 17.0.4 (64-bit) from FileHorse.com

4.IDE

Eclipse IDE for Enterprise Java

Link: https://eclipseide.org/

5.sourceTree

https://www.sourcetreeapp.com/

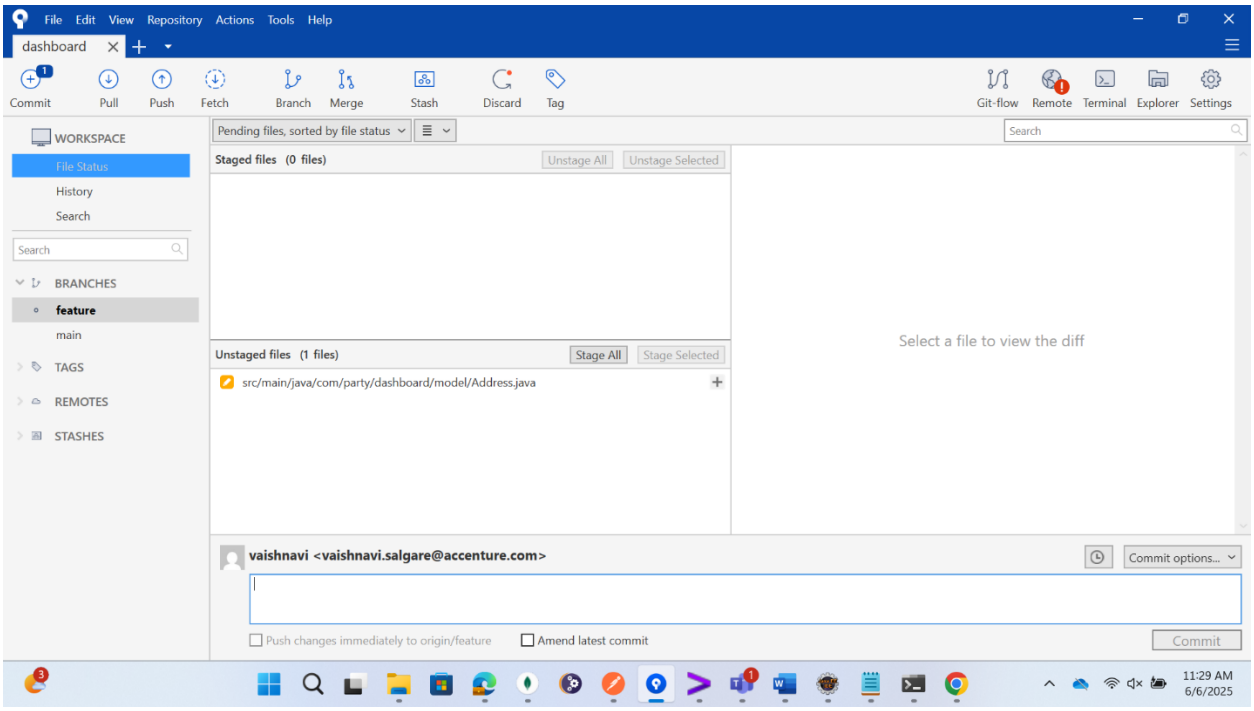
-Initialize a repo with .gitignore

-Stage files → Write message → Click Commit

-Push to GitHub as backup or for team collaboration

-Pull to GitHub

-merge to GitHub

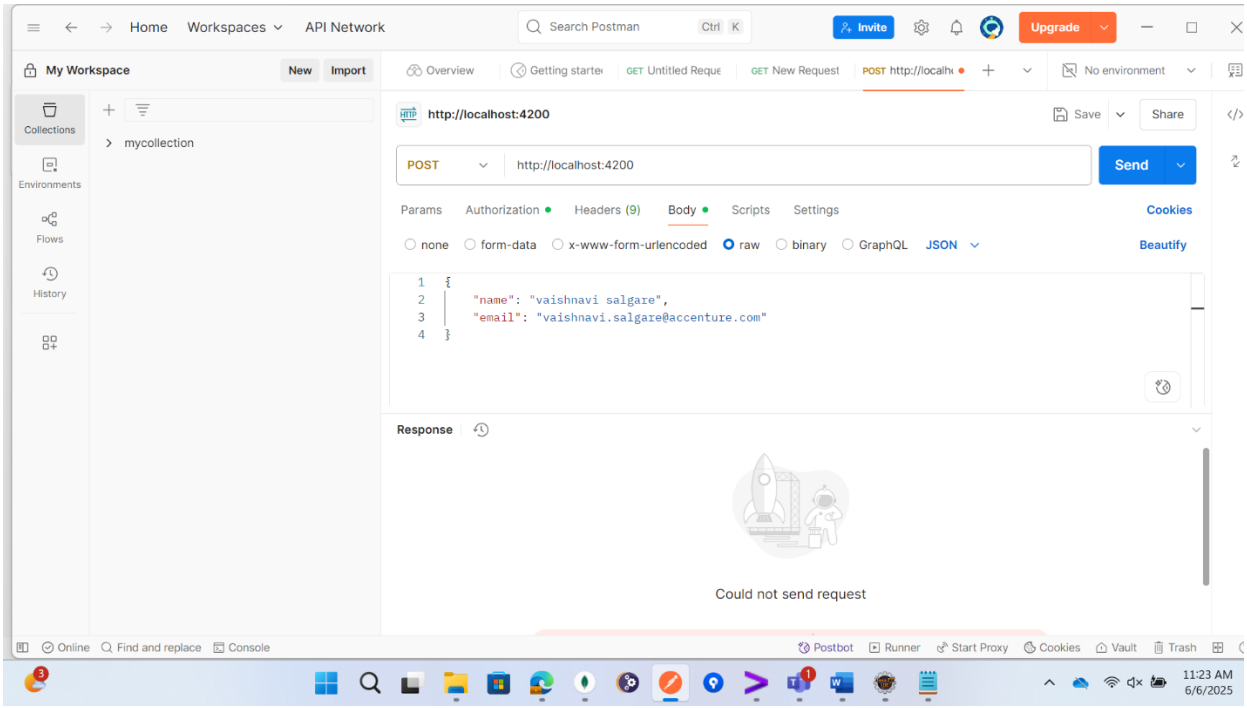


- **Tools**

1. **Git & GitHub**
   https://git-scm.com/downloads

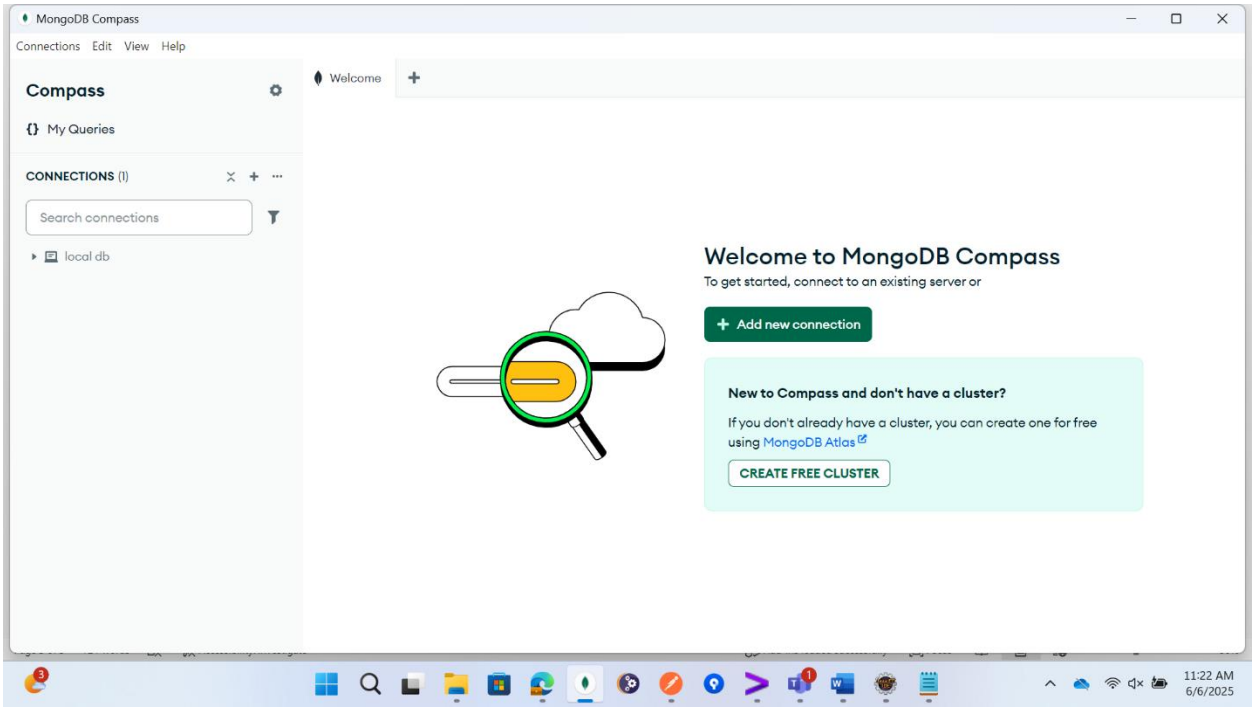
2. **Postman**
   -Testing tool

   https://www.postman.com/downloads/



3. Mongodb

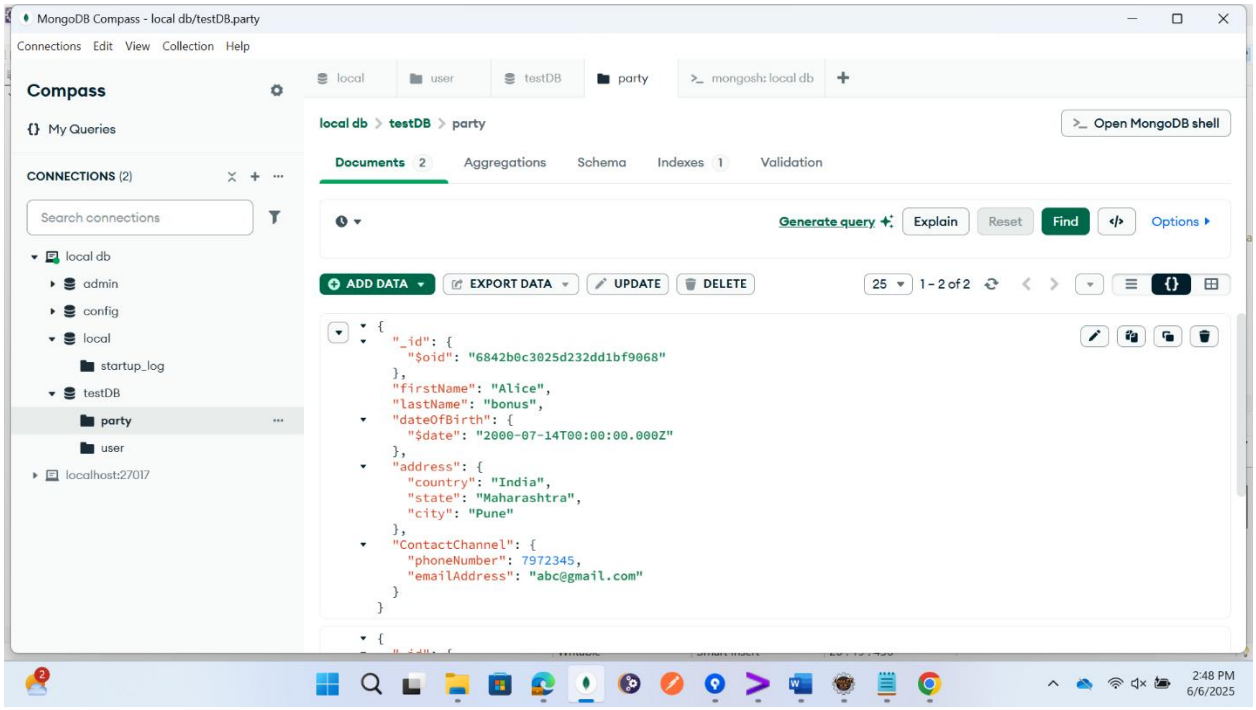
Mongodb installer:
https://www.mongodb.com/try/download/community
Mongodb GUI:https://www.mongodb.com/try/download/compass

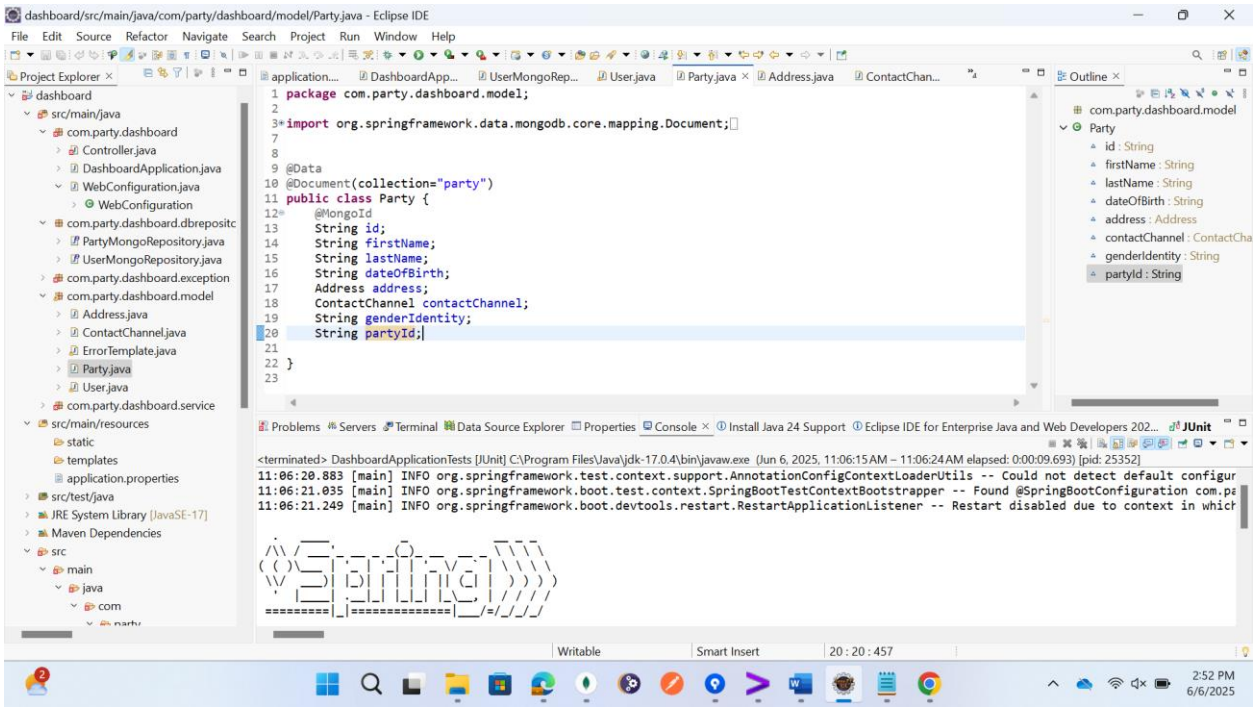


- **Created New Database**

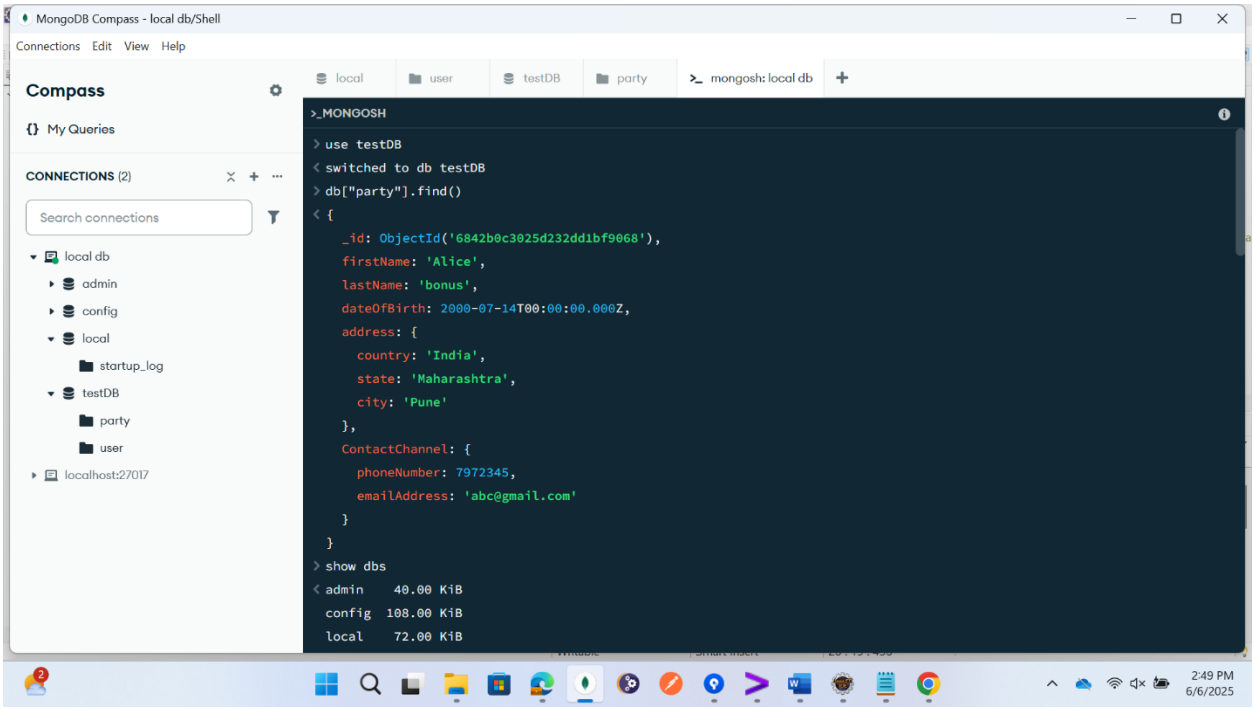
  Database name: party

  Inserted records from party.java :

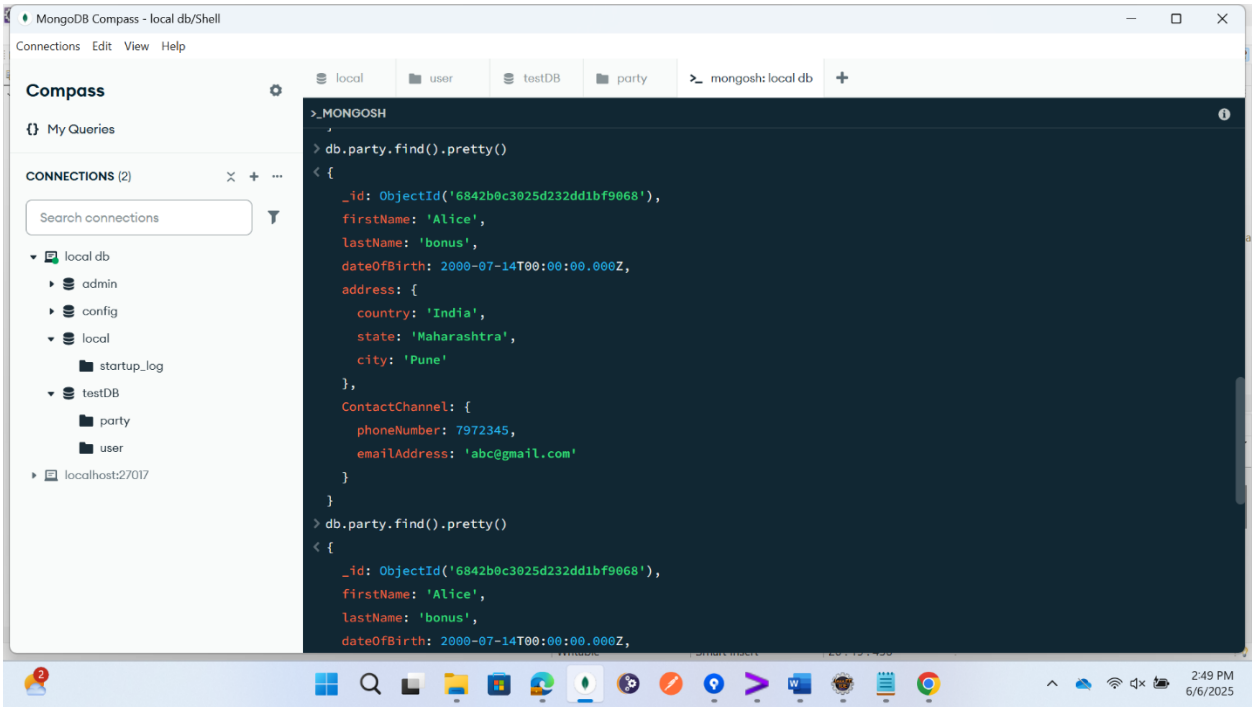




- **MongoDB Shell:**
  -db[“party”].find()



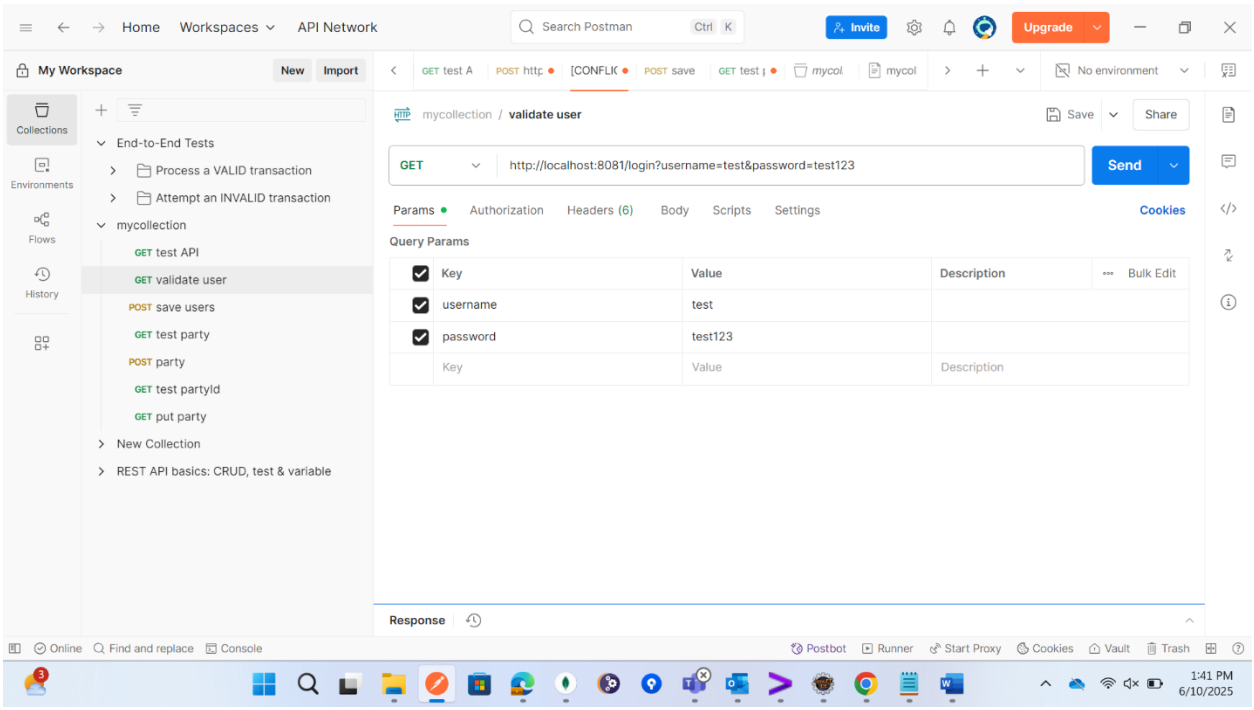
- db.party.find().pretty():
  -Proper formatted JSON

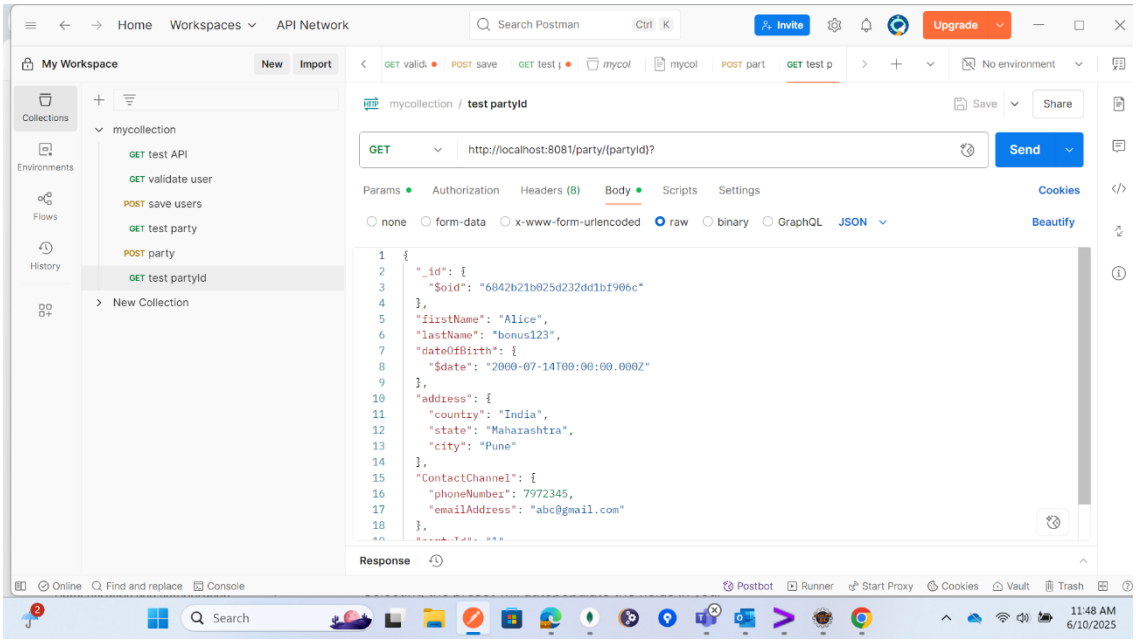


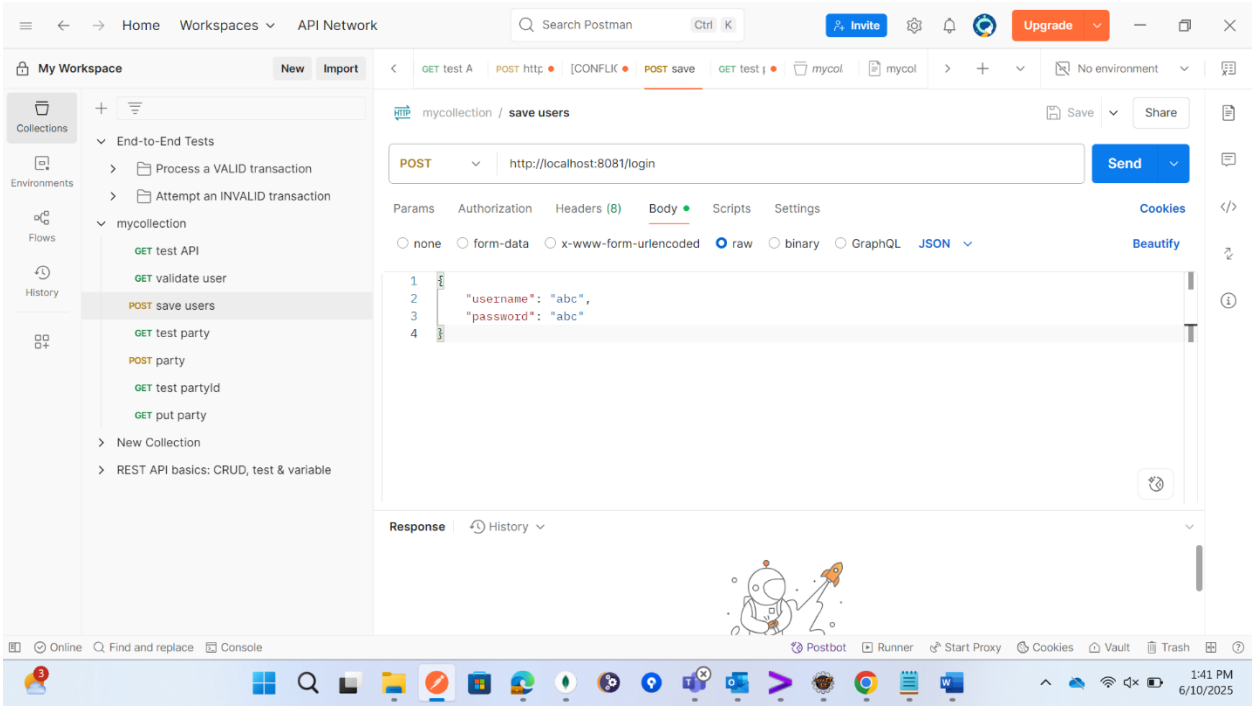
❖ API Testing (Postman):

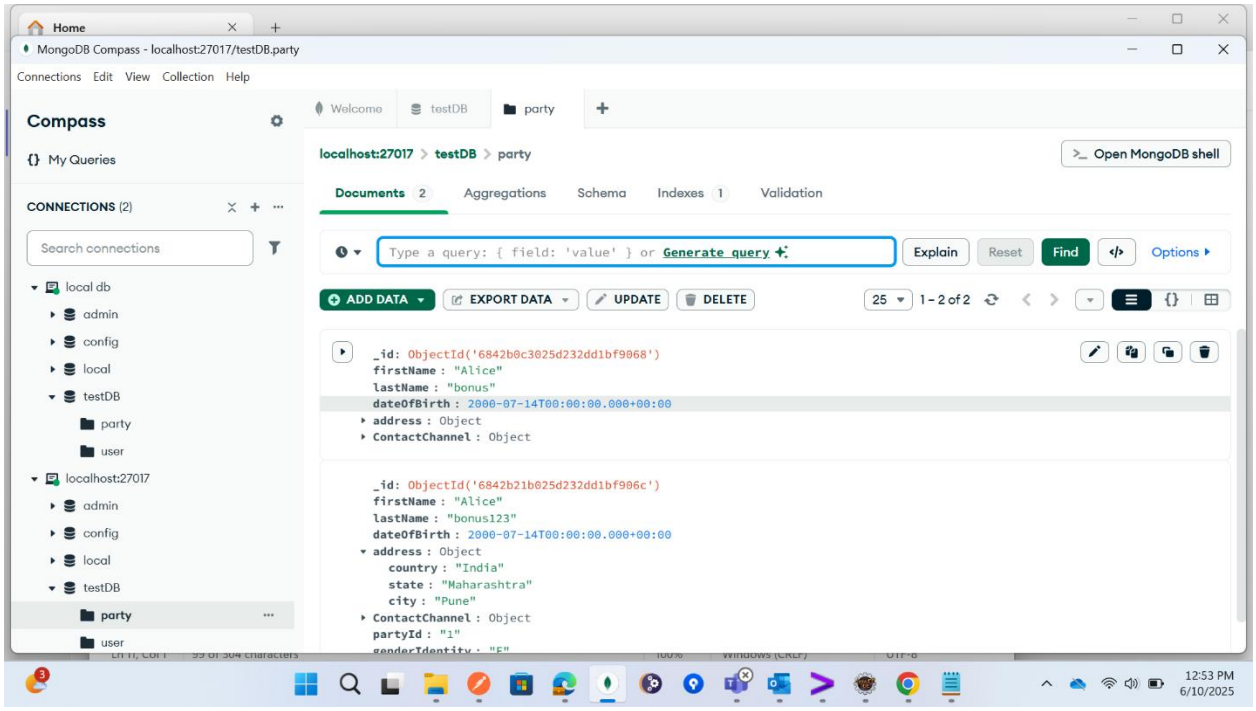
1. **HTTP Request:**

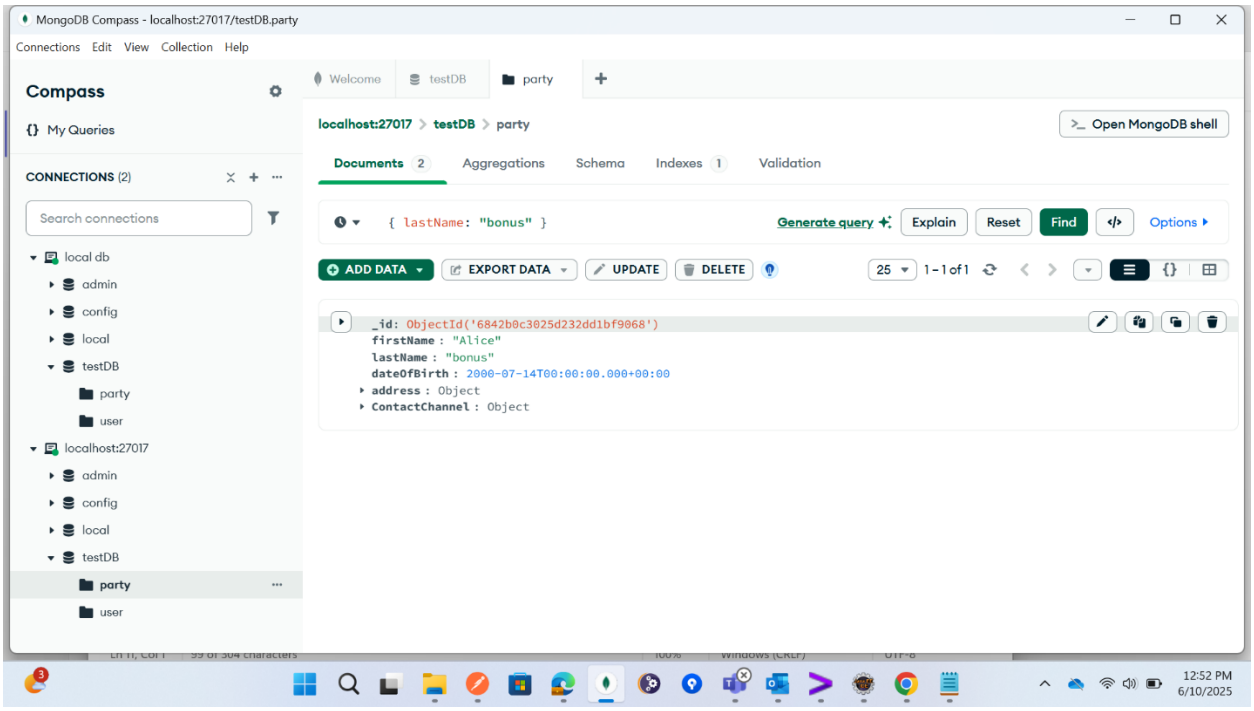
- **GET Request:** To retrieve or fetch data
- **POST Request:** To create and update data
- **PUT Request;** To update data
- **DELETE Request:** For deleting data

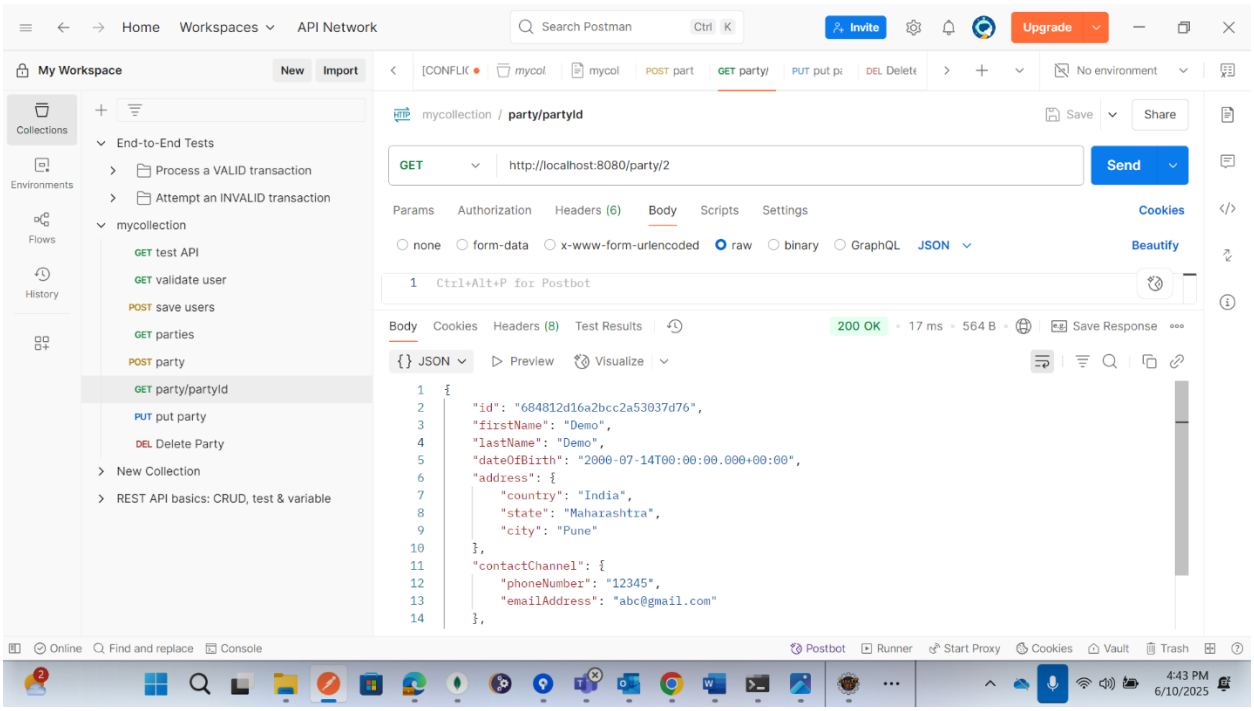




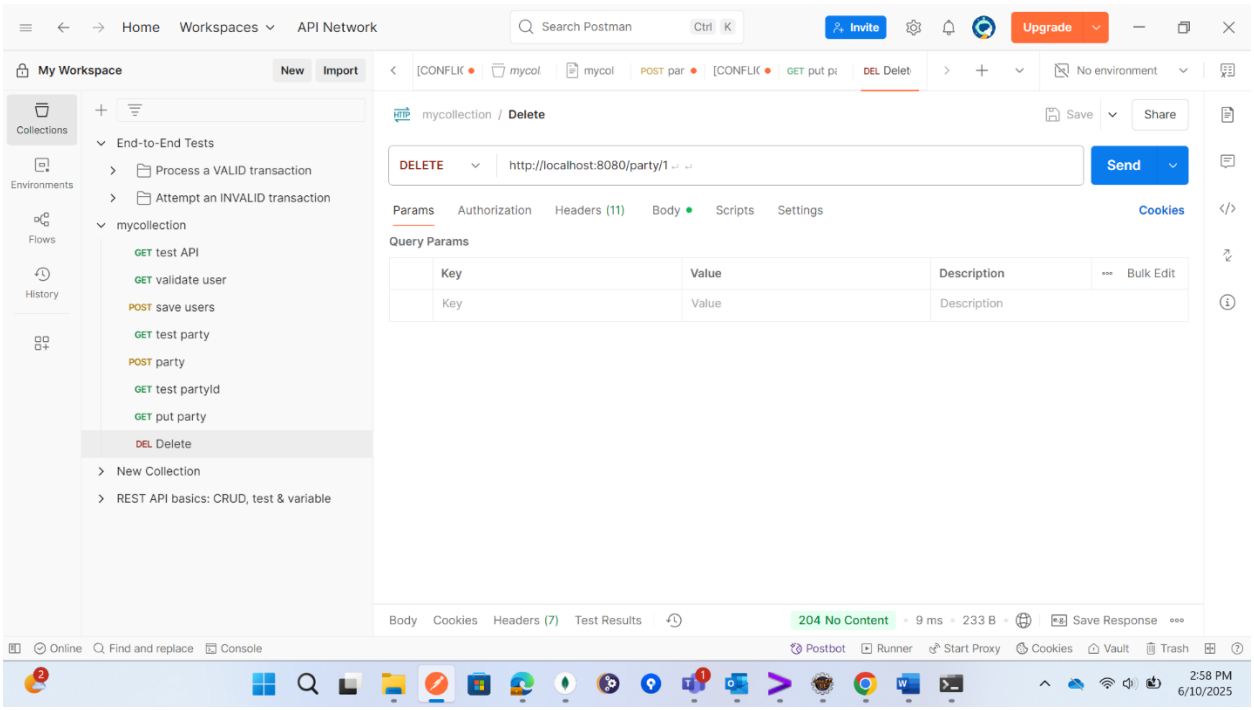

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
Flows
History
End-to-End Tests
Process a VALID transaction
Attempt an INVALID transaction
mycollection
GET test API
GET validate user
POST save users
GET test party
POST party
GET test partyId
GET put party
New Collection
REST API basics: CRUD, test & variable
mycollection / save users
Save
Share
POST
http://localhost:8081/login
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"username": "abc",
"password": "abc"
}
Response
History
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
1:41 PM
6/10/2025


Home
MongoDB Compass - localhost:27017/testDB.party
Connections
Edit
View
Collection
Help
Compass
My Queries
CONNECTIONS (2)
Search connections
local db
admin
config
local
testDB
party
user
localhost:27017
admin
config
local
testDB
party
user
Welcome
testDB
party
localhost:27017 > testDB > party
Open MongoDB shell
Documents 2
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 2 of 2
_id: ObjectId('6842b0c3025d232dd1bf9068')
firstName : "Alice"
lastName : "bonus"
dateOfBirth : 2000-07-14T00:00:00.000+00:00
address : Object
ContactChannel : Object
_id: ObjectId('6842b21b025d232dd1bf906c')
firstName : "Alice"
lastName : "bonus123"
dateOfBirth : 2000-07-14T00:00:00.000+00:00
address : Object
country : "India"
state : "Maharashtra"
city : "Pune"
ContactChannel : Object
partyId : "1"
12:53 PM
6/10/2025


MongoDB Compass - localhost:27017/testDB.party
Connections Edit View Collection Help
Compass
My Queries
CONNECTIONS (2)
Search connections
local db
admin
config
local
testDB
party
user
localhost:27017
admin
config
local
testDB
party
user
Welcome
testDB
party
localhost:27017 > testDB > party
Open MongoDB shell
Documents 2
Aggregations
Schema
Indexes 1
Validation
{ lastName: "bonus" }
Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 1 of 1
_id: ObjectId('6842b0c3025d232dd1bf9068')
firstName : "Alice"
lastName : "bonus"
dateOfBirth : 2000-07-14T00:00:00.000+00:00
address : Object
ContactChannel : Object
12:52 PM
6/10/2025

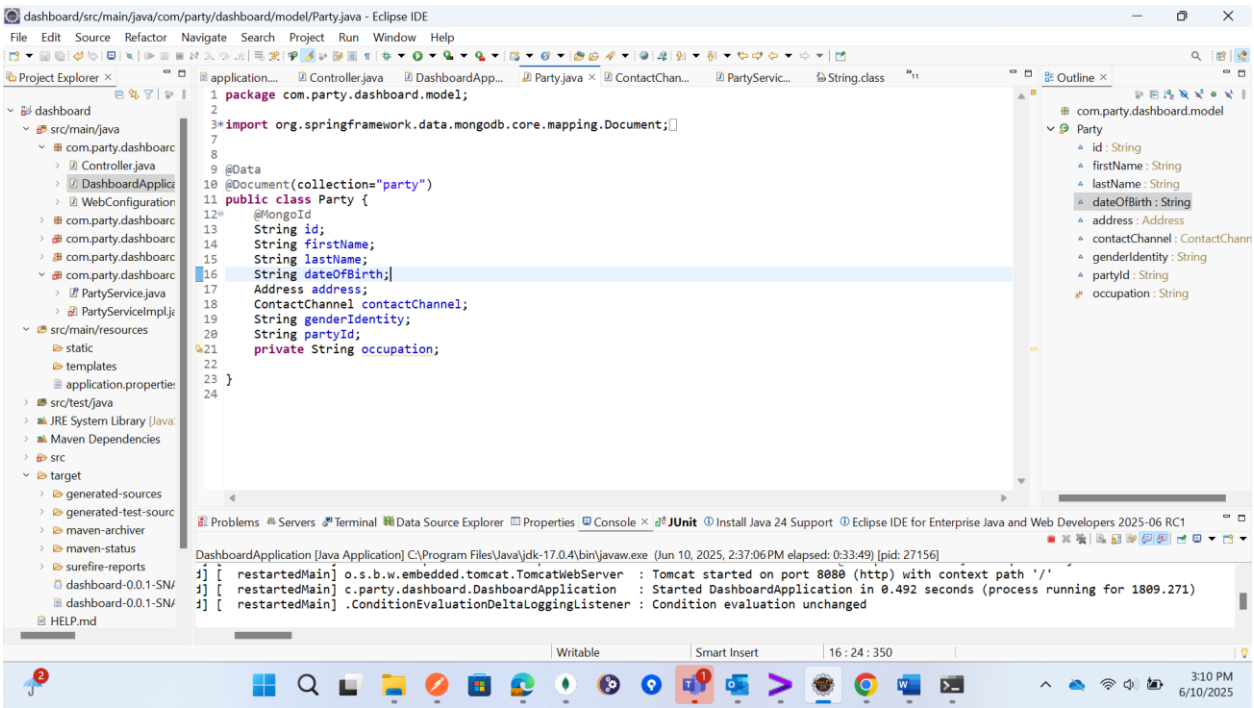

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
Flows
History
End-to-End Tests
Process a VALID transaction
Attempt an INVALID transaction
mycollection
GET test API
GET validate user
POST save users
GET parties
POST party
GET party/partyId
PUT put party
DEL Delete Party
New Collection
REST API basics: CRUD, test & variable
mycollection / party/partyId
Save
Share
GET
http://localhost:8080/party/2
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
Ctrl+Alt+P for Postbot
Body
Cookies
Headers (8)
Test Results
200 OK
17 ms
564 B
Save Response
JSON
Preview
Visualize
{
"id": "684812d16a2bcc2a53037d76",
"firstName": "Demo",
"lastName": "Demo",
"dateOfBirth": "2000-07-14T00:00:00.000+00:00",
"address": {
"country": "India",
"state": "Maharashtra",
"city": "Pune"
},
"contactChannel": {
"phoneNumber": "12345",
"emailAddress": "abc@gmail.com"
},
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
4:43 PM
6/10/2025

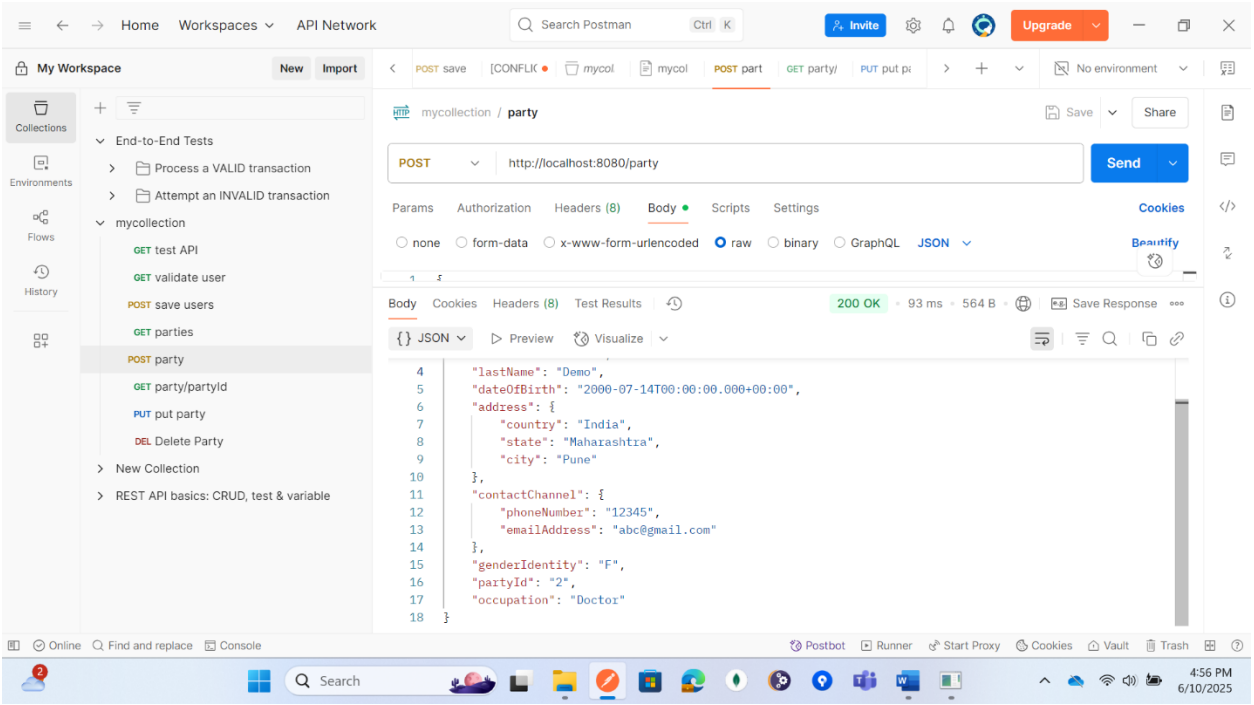
- Delete:

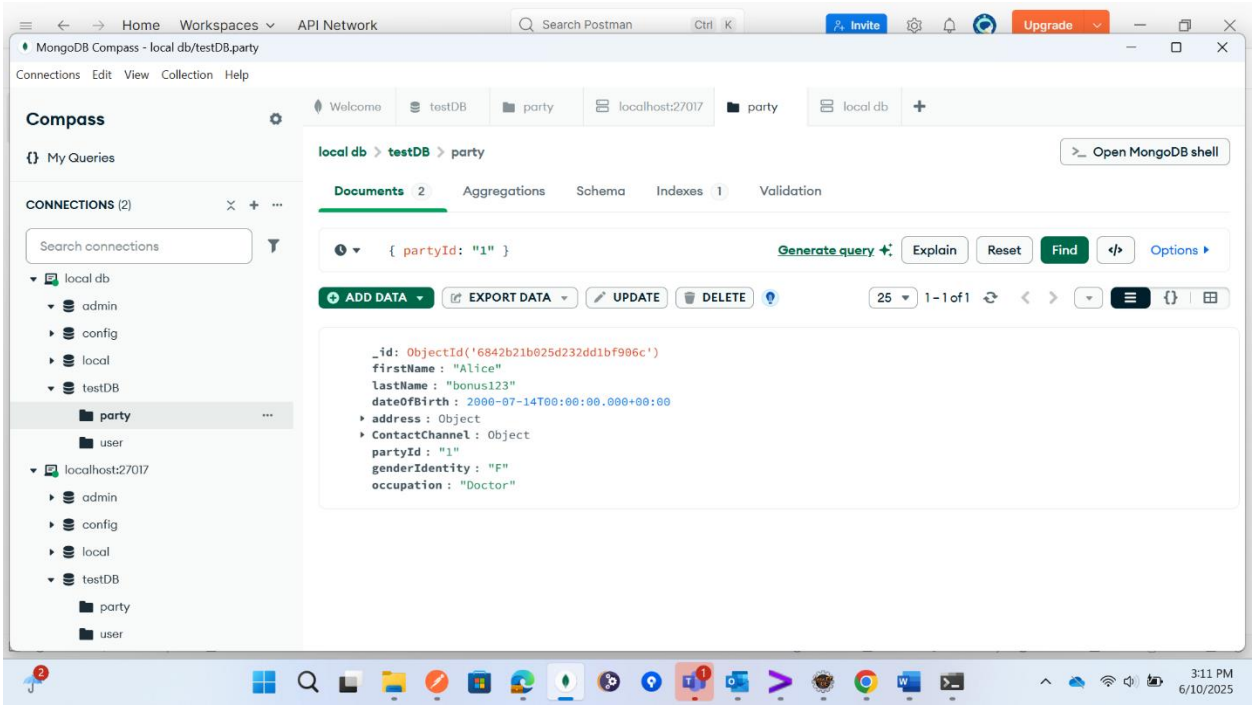


Task:

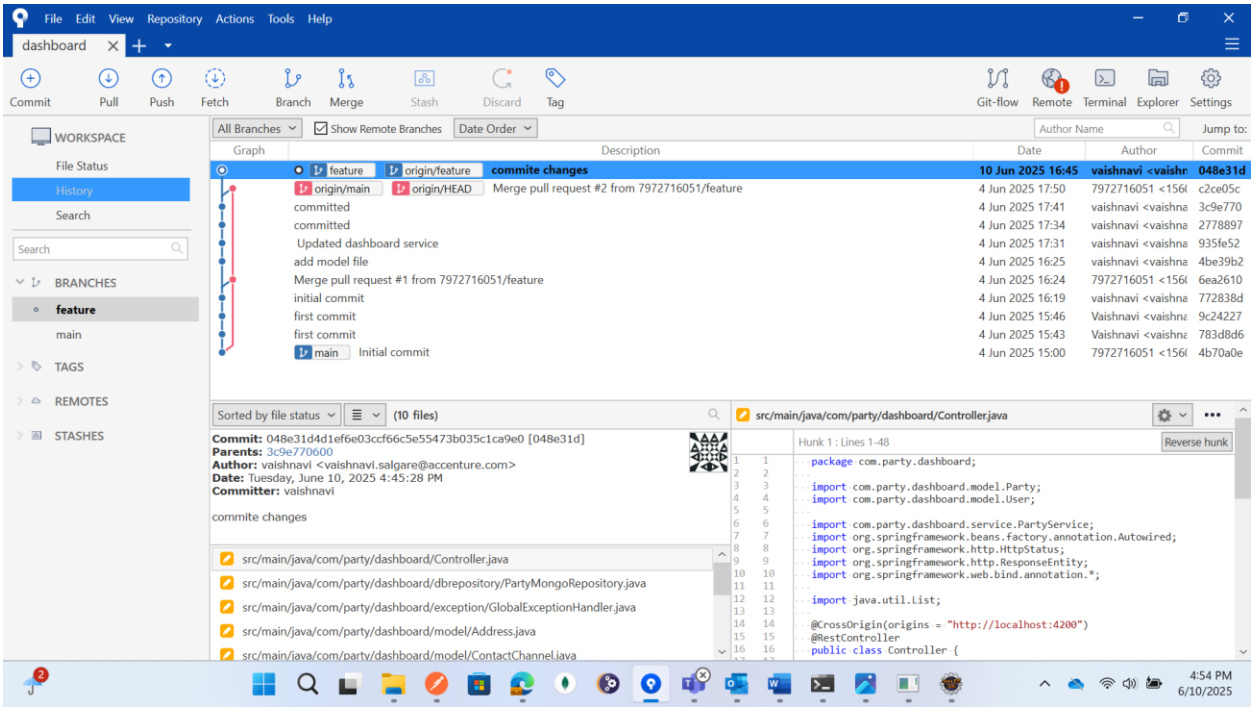
Adding occupation field in Party model class then Save API in occupation field add data in DB Apply same for get and update.

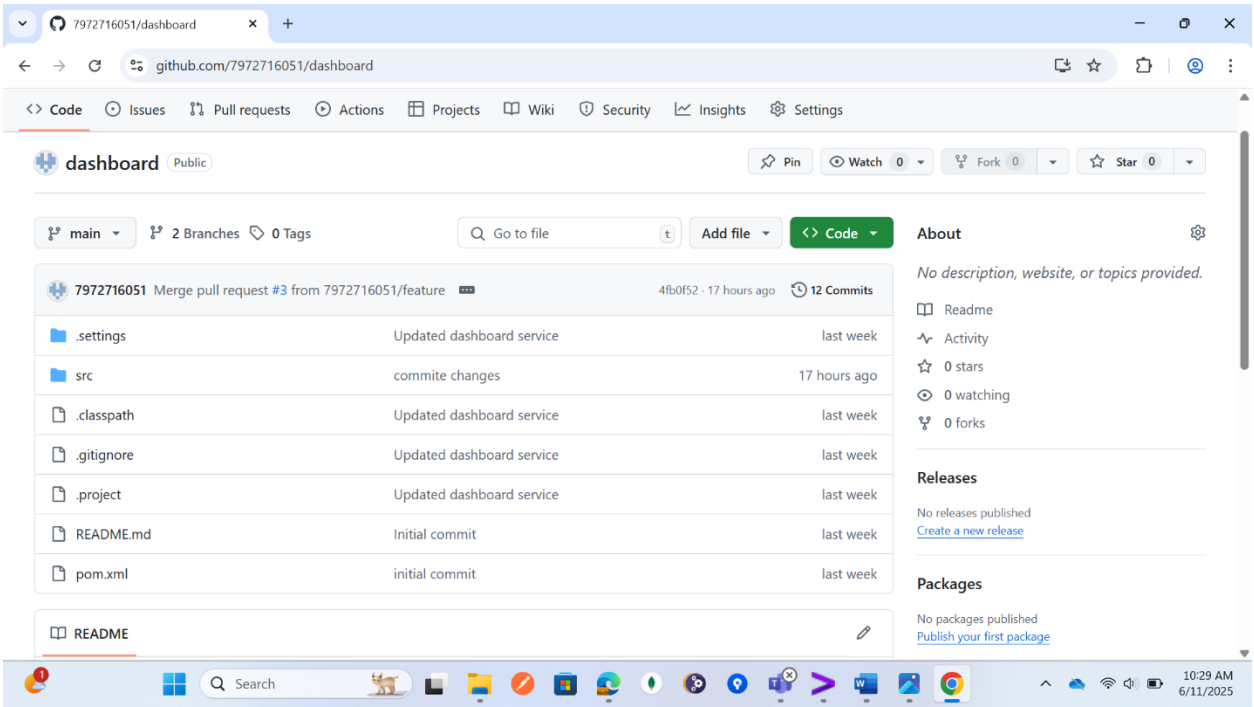



Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
Flows
History
End-to-End Tests
Process a VALID transaction
Attempt an INVALID transaction
mycollection
GET test API
GET validate user
POST save users
GET parties
POST party
GET party/partyId
PUT put party
DEL Delete Party
New Collection
REST API basics: CRUD, test & variable
mycollection / party
Save
Share
POST
http://localhost:8080/party
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
Body
Cookies
Headers (8)
Test Results
200 OK
93 ms
564 B
Save Response
JSON
Preview
Visualize
{
"lastName": "Demo",
"dateOfBirth": "2000-07-14T00:00:00.000+00:00",
"address": {
"country": "India",
"state": "Maharashtra",
"city": "Pune"
},
"contactChannel": {
"phoneNumber": "12345",
"emailAddress": "abc@gmail.com"
},
"genderIdentity": "F",
"partyId": "2",
"occupation": "Doctor"
}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
Search
4:56 PM
6/10/2025

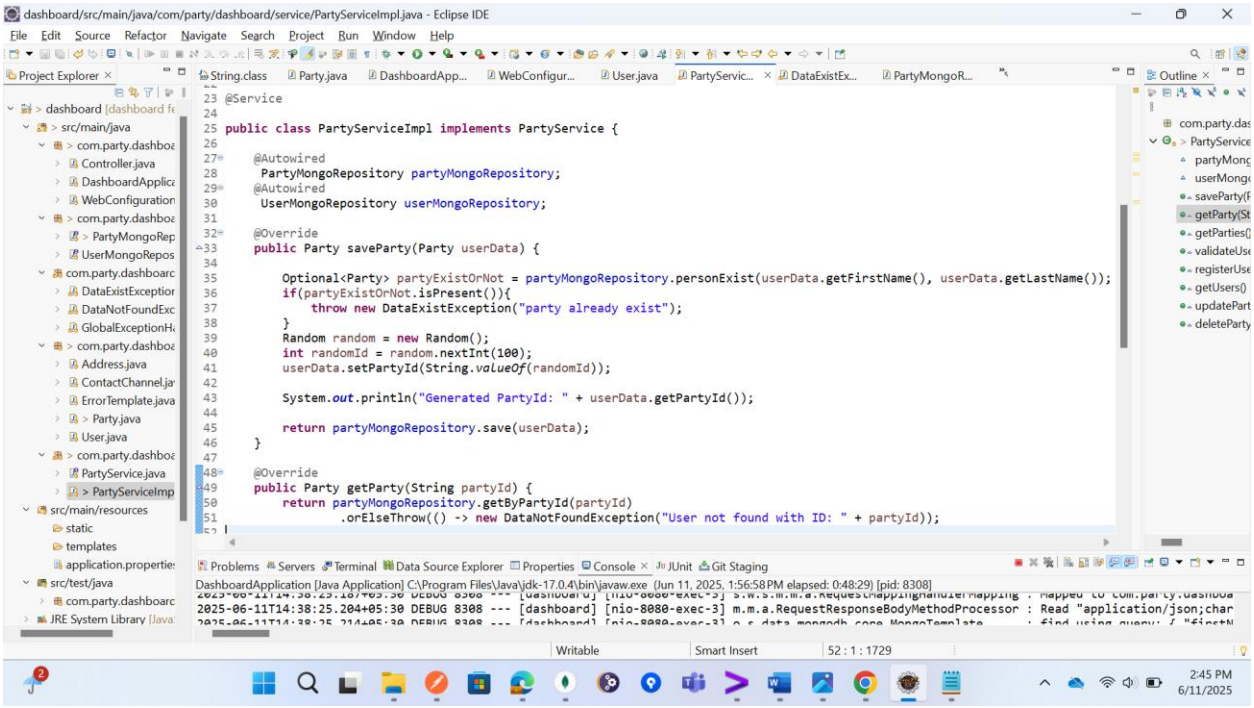

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
MongoDB Compass - local db/testDB.party
Connections
Edit
View
Collection
Help
Compass
My Queries
CONNECTIONS (2)
Search connections
local db
admin
config
local
testDB
party
user
localhost:27017
admin
config
local
testDB
party
user
Welcome
testDB
party
localhost:27017
party
local db
local db > testDB > party
Open MongoDB shell
Documents 2
Aggregations
Schema
Indexes 1
Validation
{ partyId: "1" }
Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 1 of 1
_id: ObjectId('6842b21b025d232dd1bf906c')
firstName : "Alice"
lastName : "bonus123"
dateOfBirth : 2000-07-14T00:00:00.000+00:00
address : Object
ContactChannel : Object
partyId : "1"
genderIdentity : "F"
occupation : "Doctor"
3:11 PM
6/10/2025

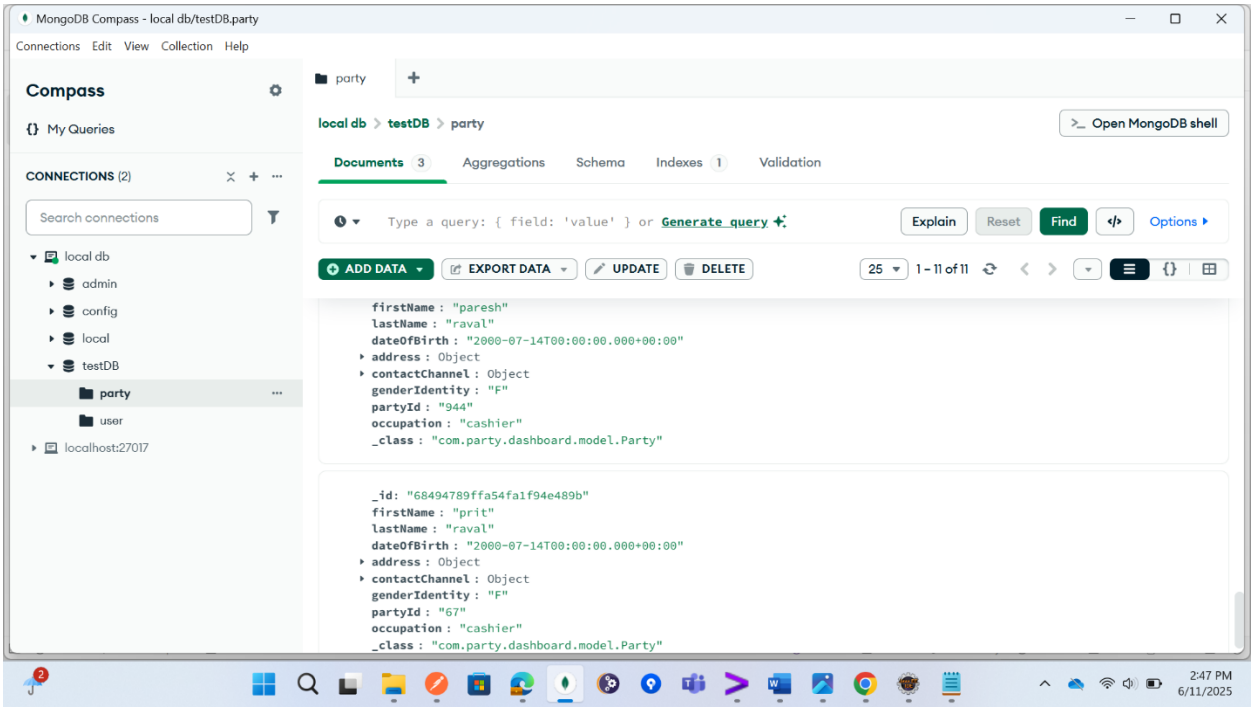
- Push , Pull and merged files into Git:

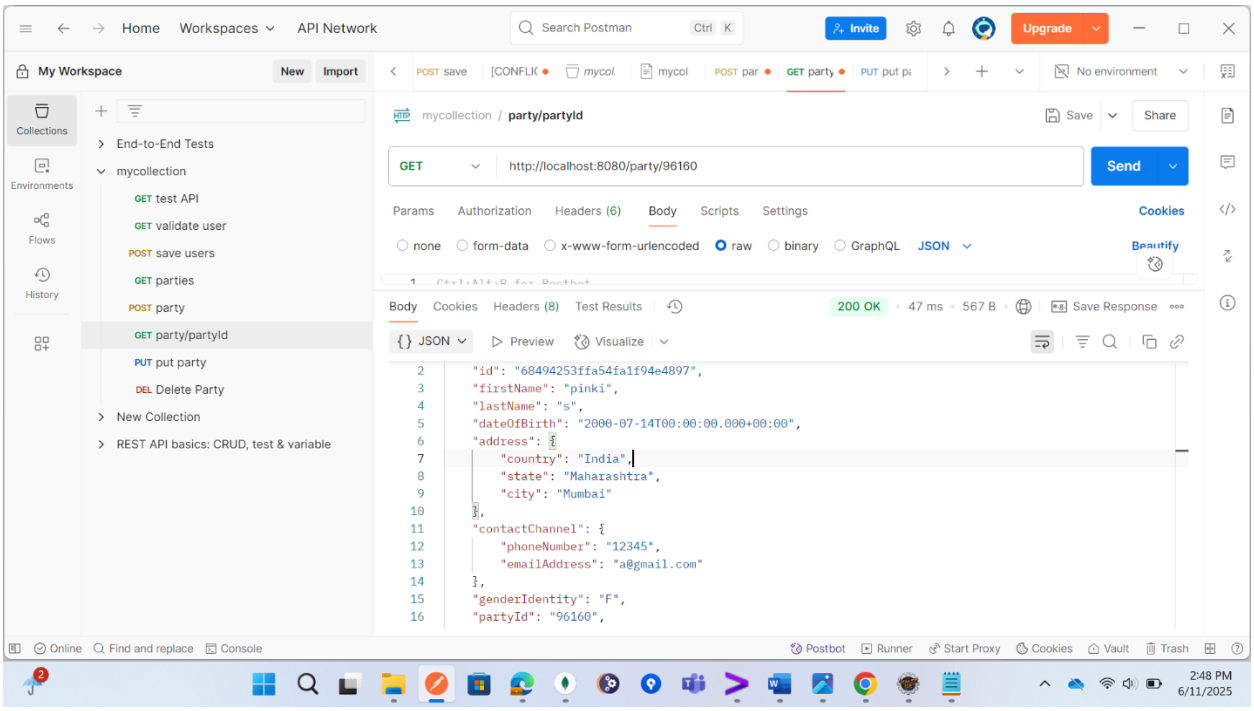




- Auto generated PartyId for users:



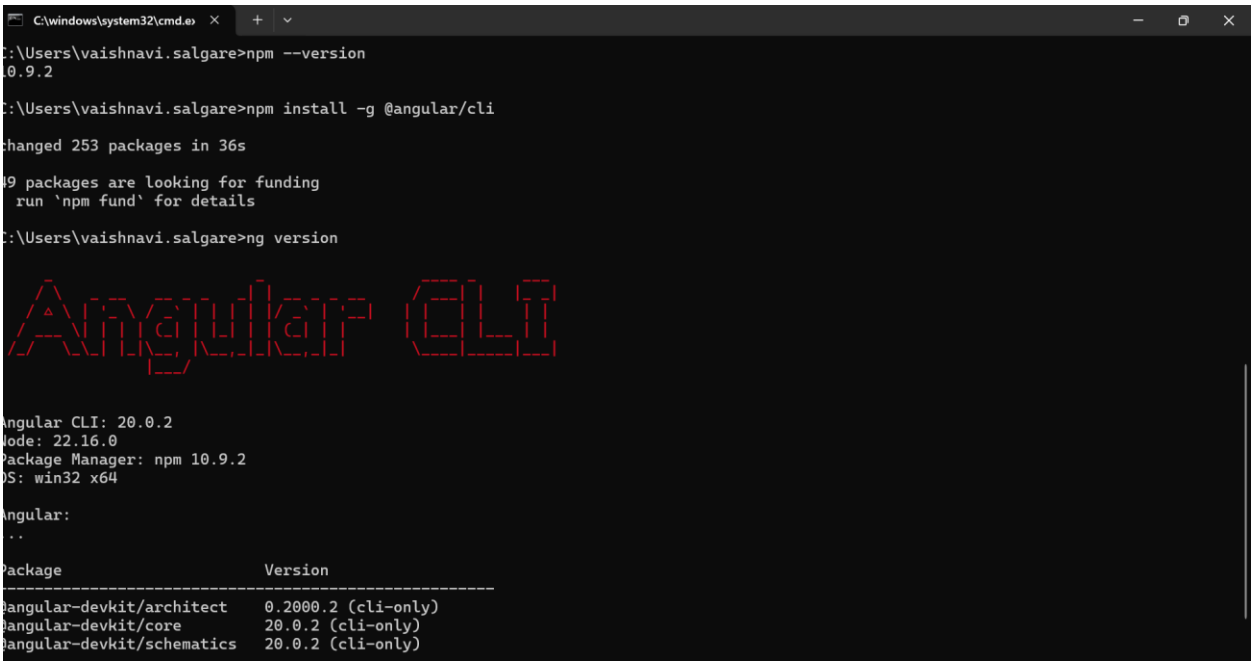


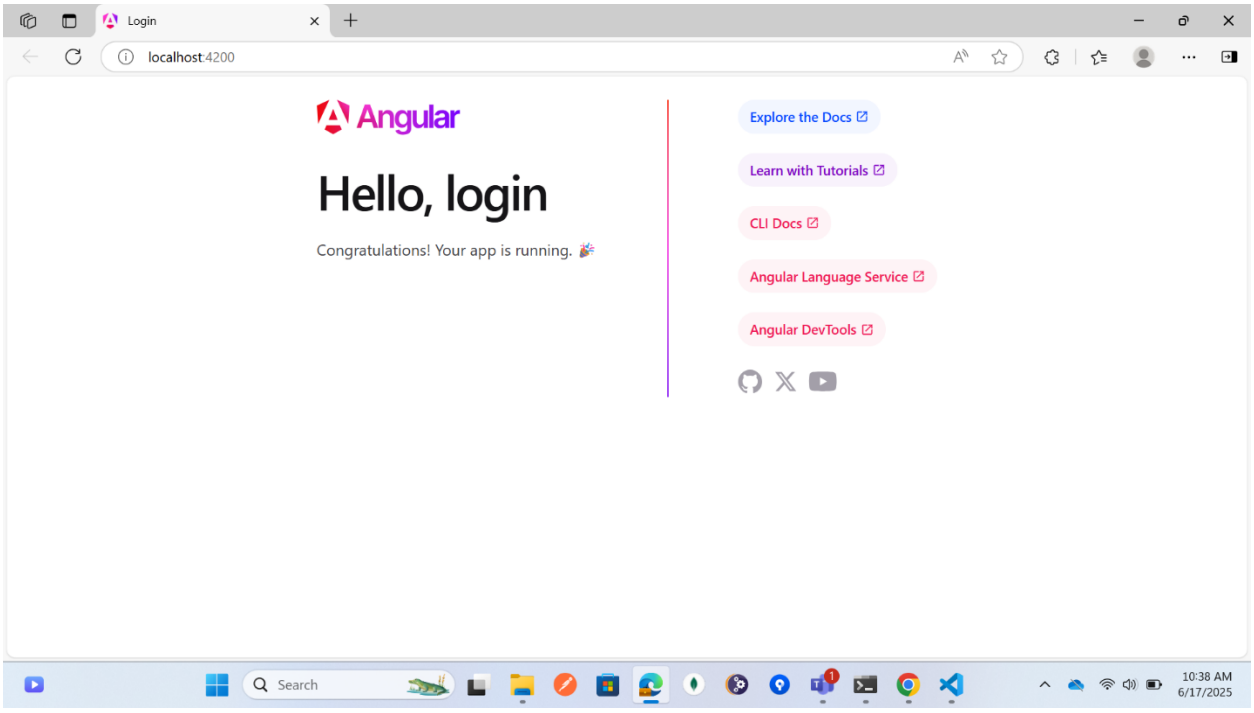

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
Flows
History
End-to-End Tests
mycollection
GET test API
GET validate user
POST save users
GET parties
POST party
GET party/partyId
PUT put party
DEL Delete Party
New Collection
REST API basics: CRUD, test & variable
No environment
mycollection / party/partyId
Save
Share
GET
http://localhost:8080/party/96160
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
Body
Cookies
Headers (8)
Test Results
200 OK
47 ms
567 B
Save Response
JSON
Preview
Visualize
"id": "68494253ffa54fa1f94e4897",
"firstName": "pinki",
"lastName": "s",
"dateOfBirth": "2000-07-14T00:00:00.000+00:00",
"address": {
"country": "India",
"state": "Maharashtra",
"city": "Mumbai"
},
"contactChannel": {
"phoneNumber": "12345",
"emailAddress": "a@gmail.com"
},
"genderIdentity": "F",
"partyId": "96160",
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
2:48 PM
6/11/2025

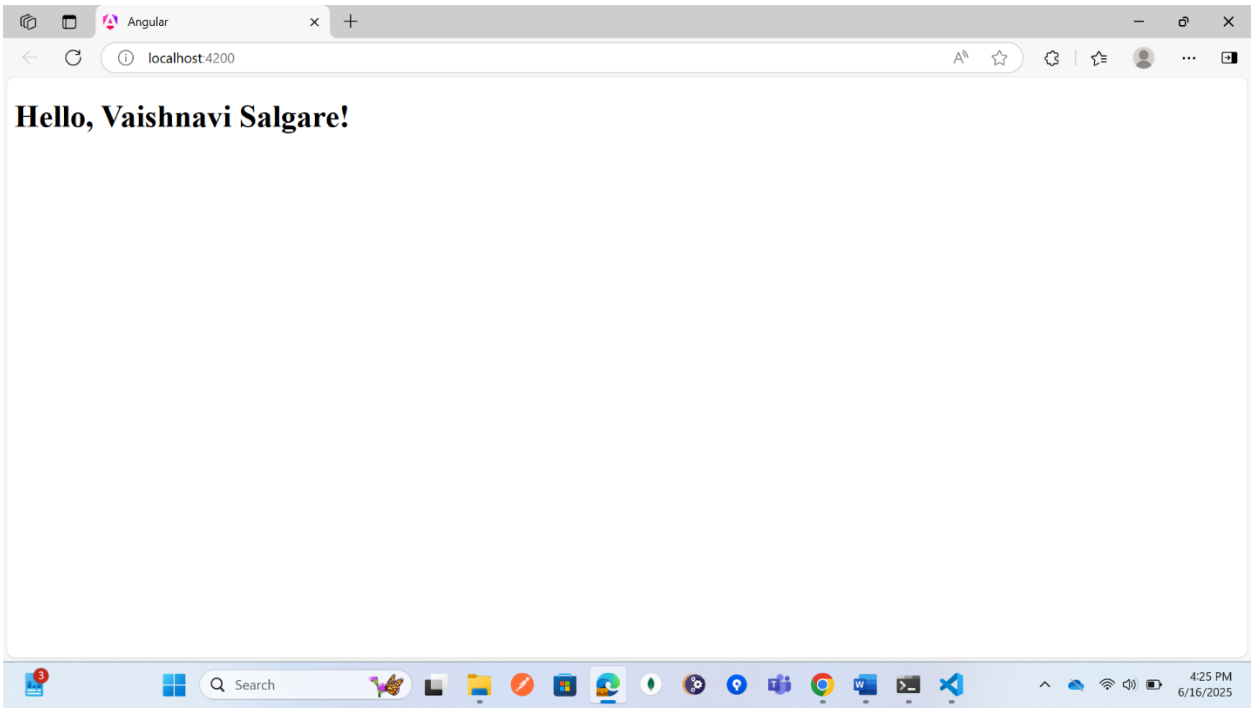
- **Frontend Technologies:**
  Angular:
- **To install Angular:**

  https://nodejs.org/

  npm install -g @angular/cli@16, #for angular install globally
  ng new appname,
  ng version,
  ng generate component(ng g c component-name) --skip-tests,
  ng generate service(ng g s service-name)

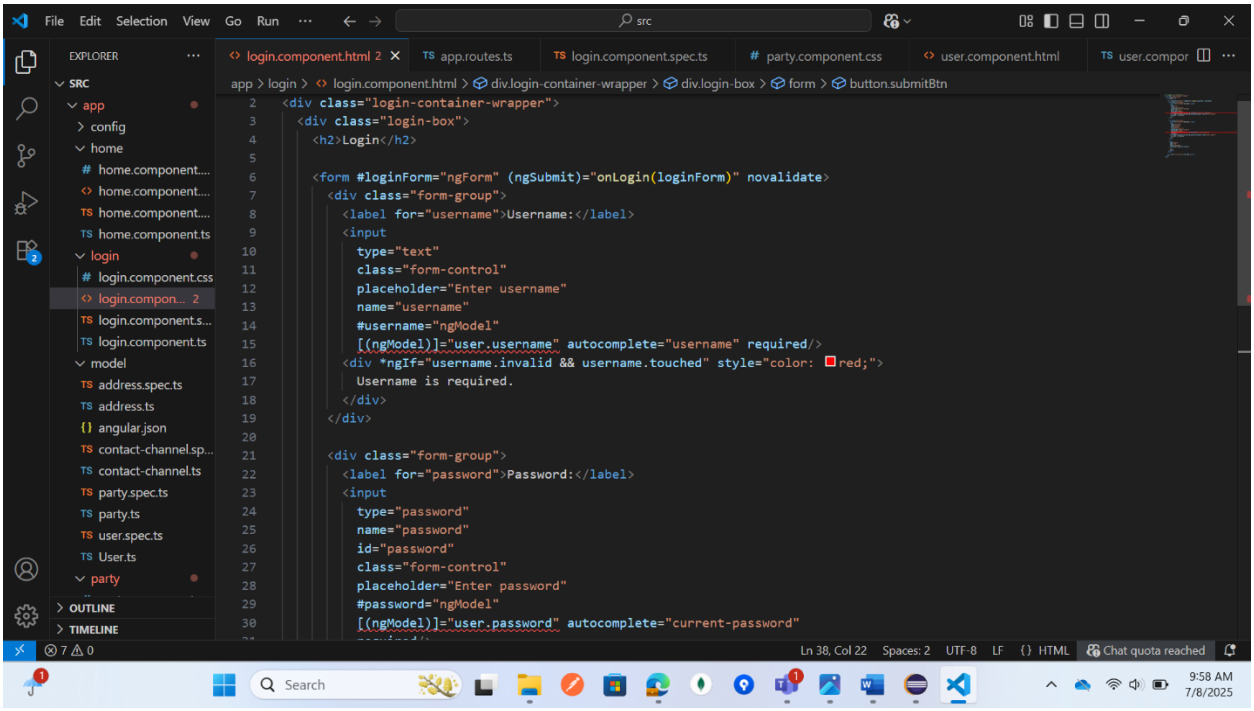




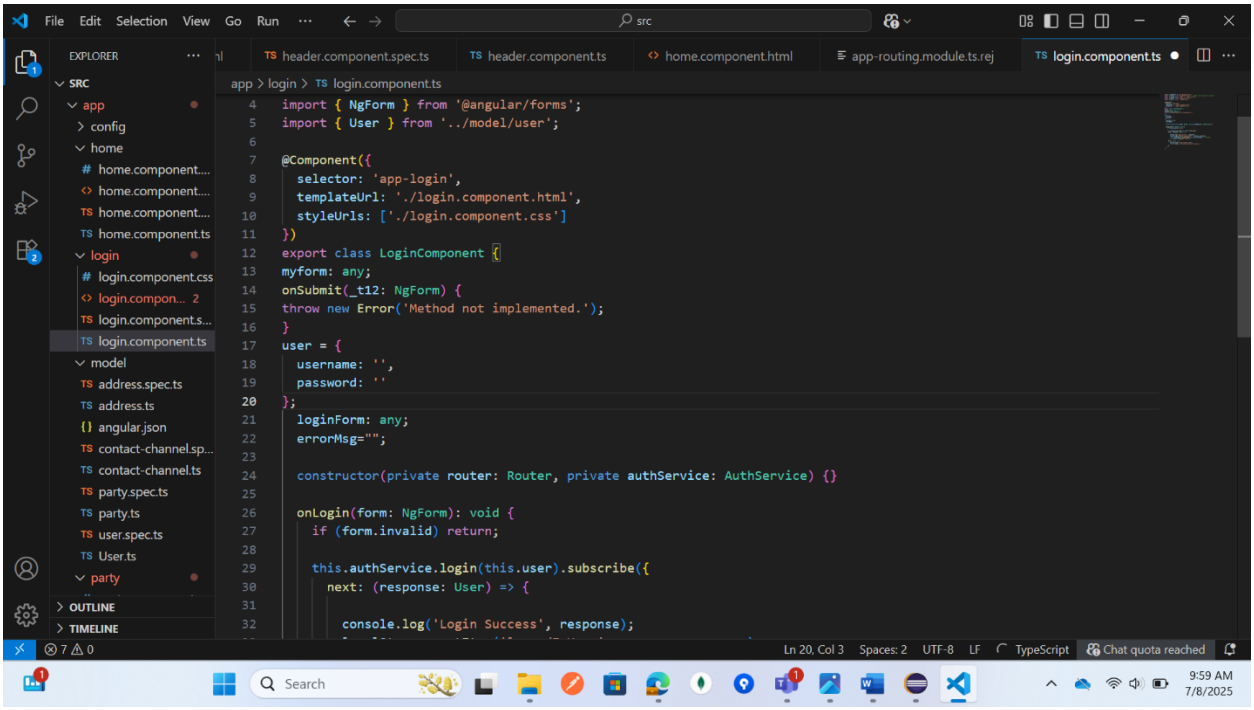


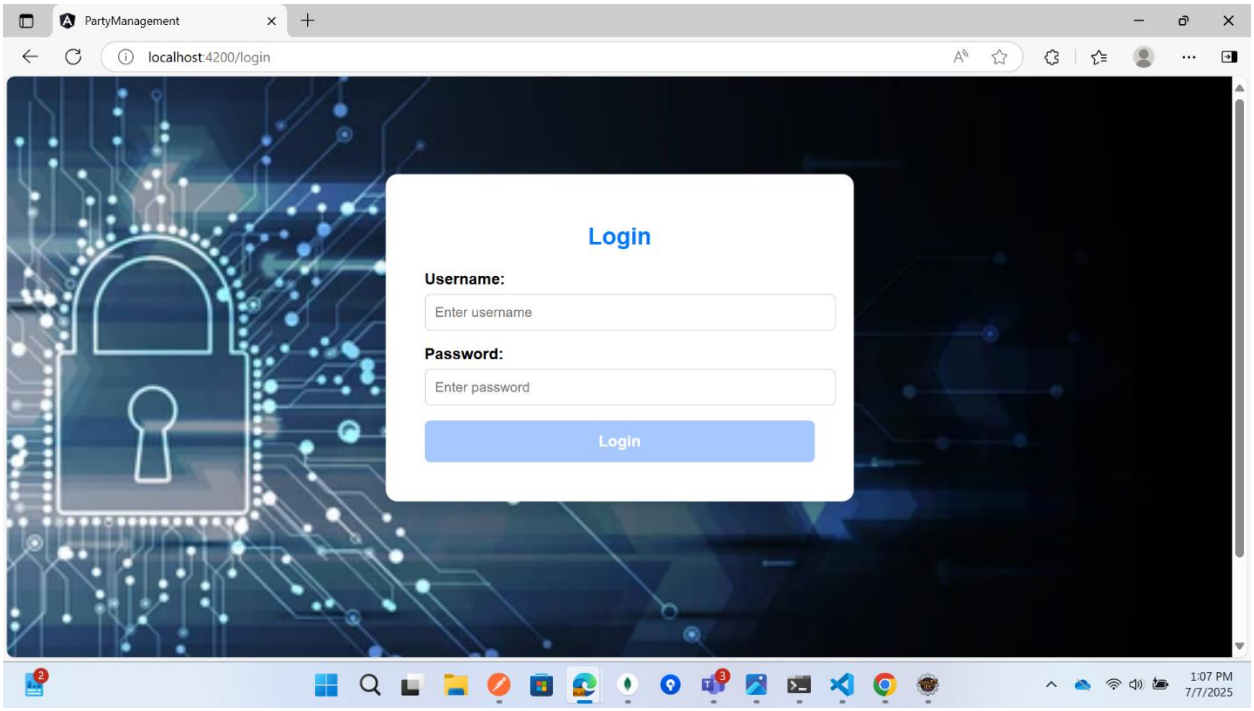
- **Creating Frontend For the same:**

Login.component.html

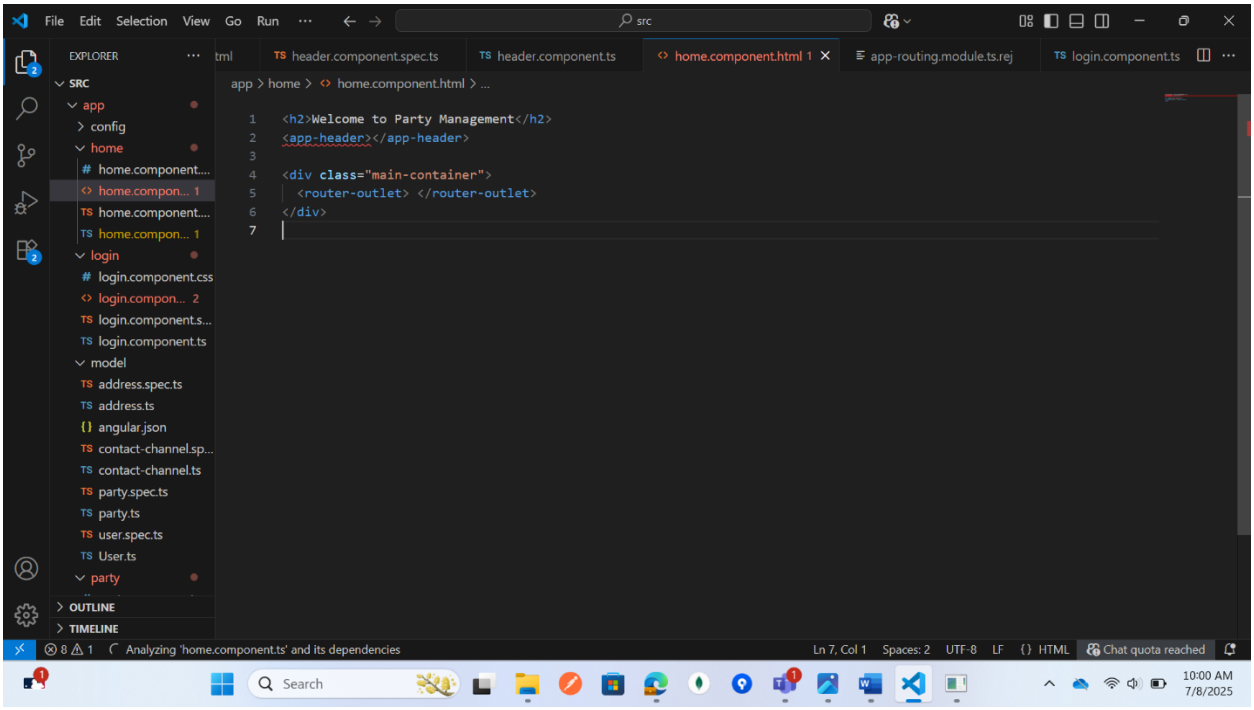


Login.component.ts

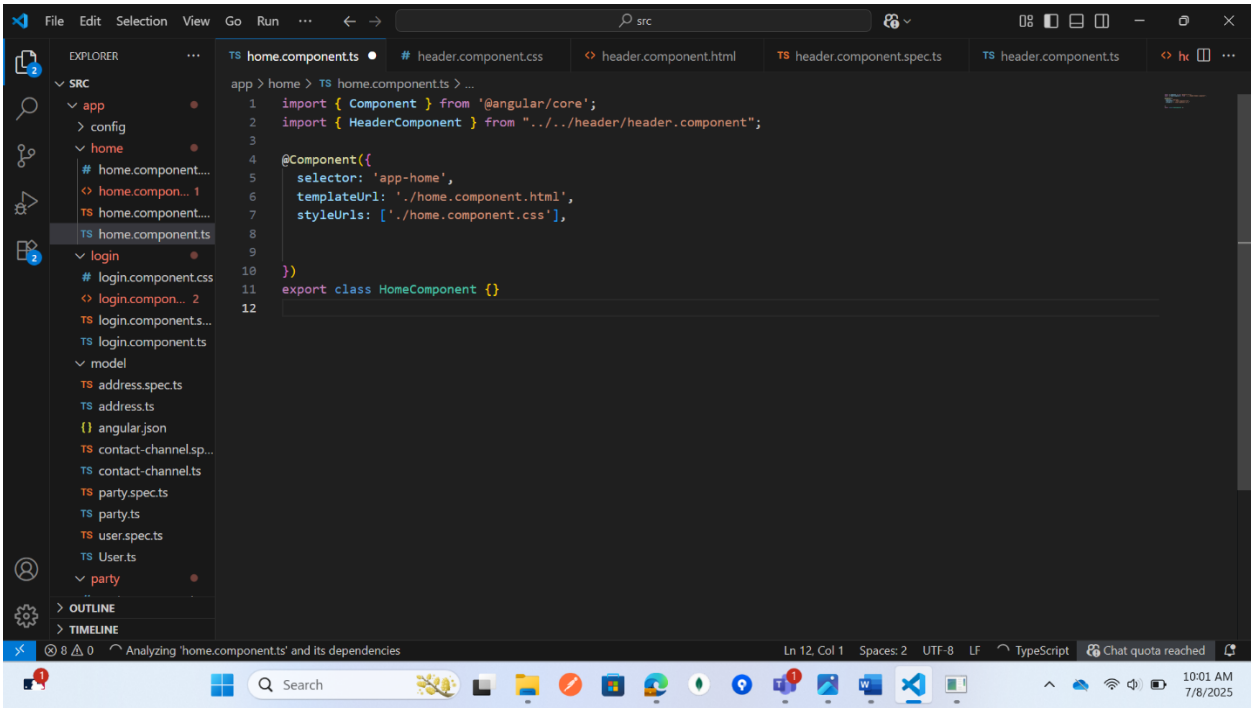


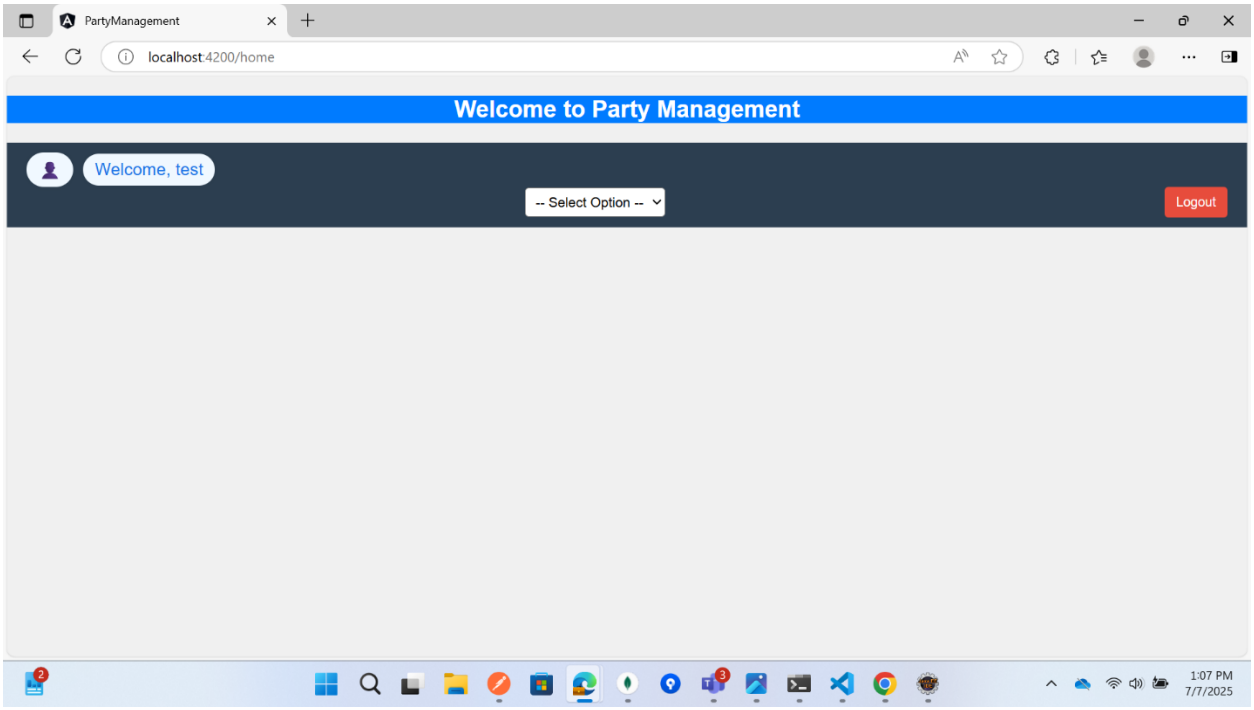


Home.component.html

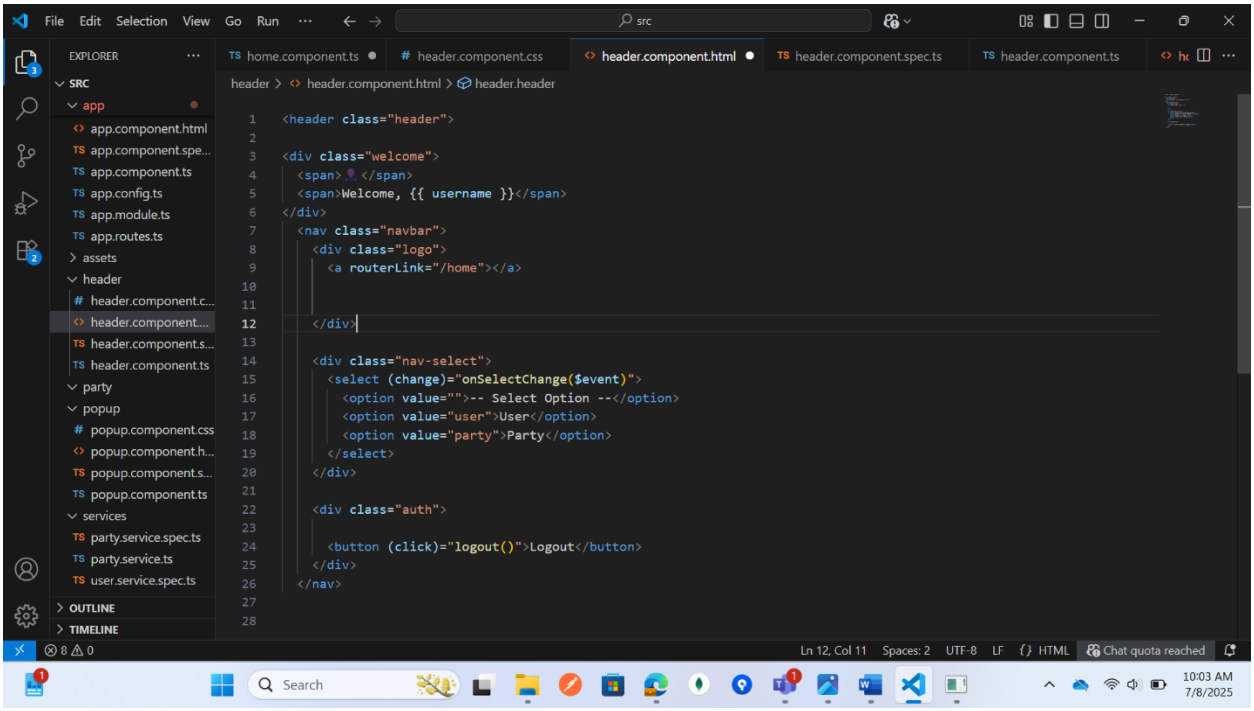


Home.component.ts

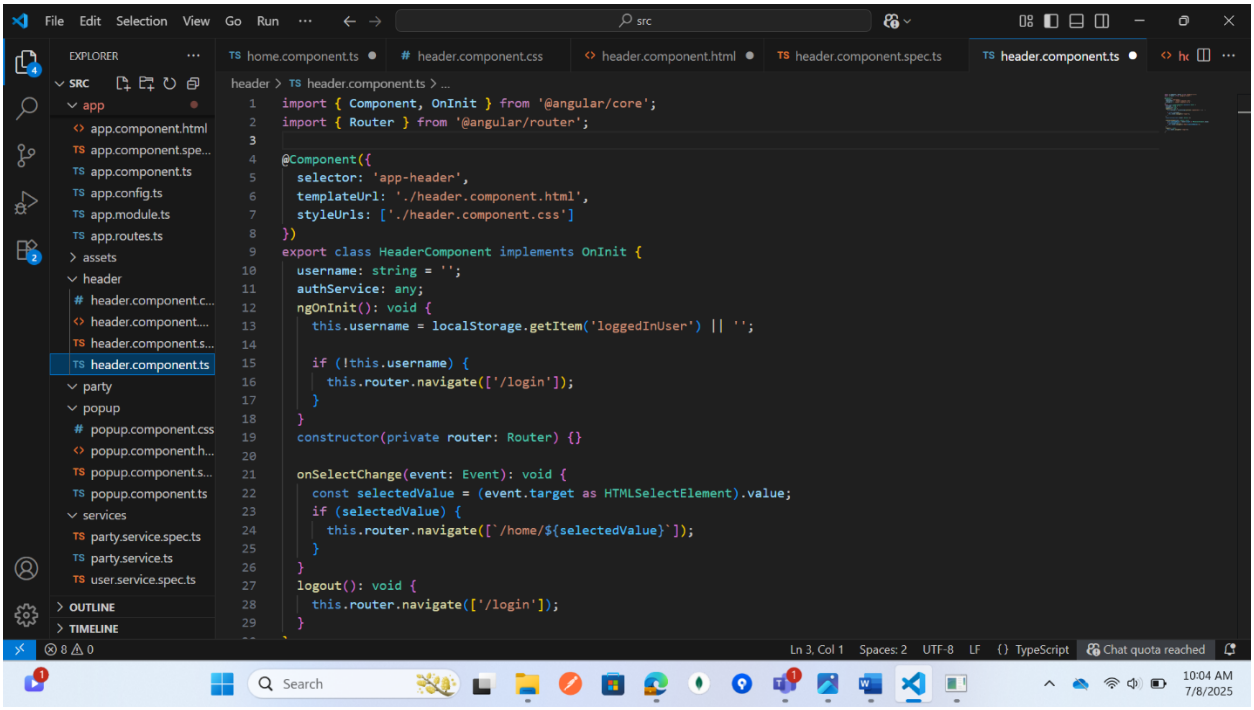


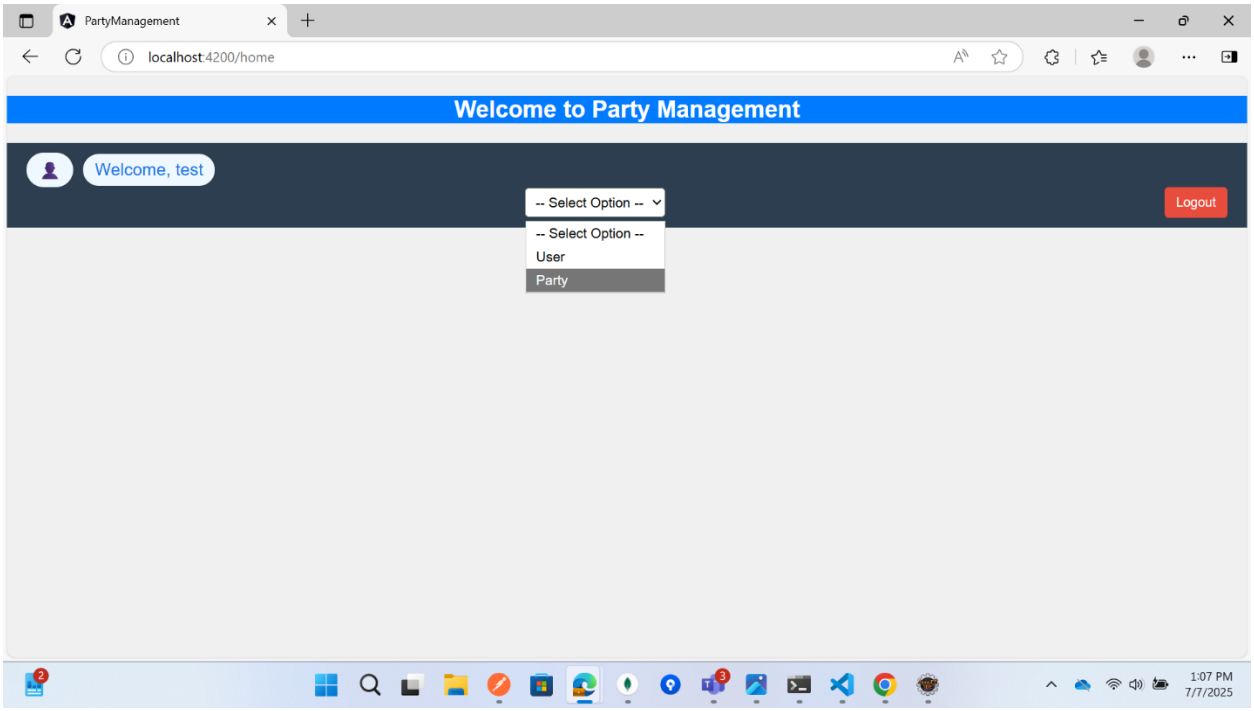


Header.component.html

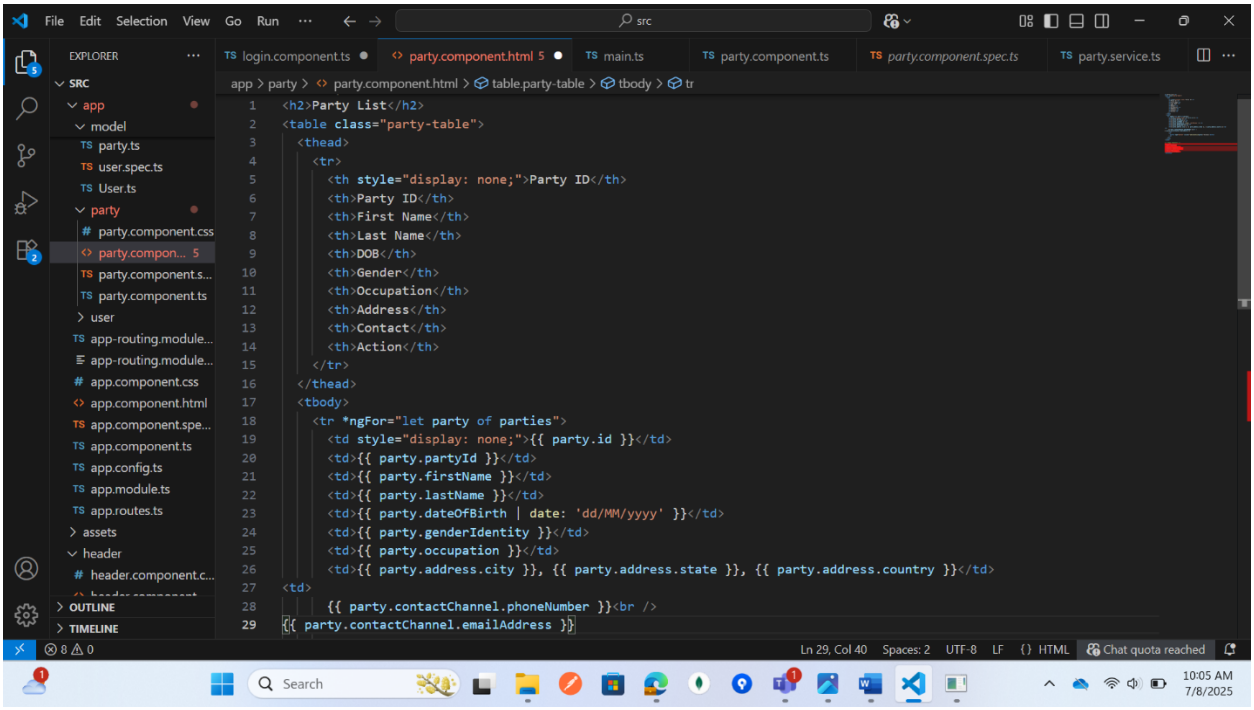


Header.component.ts

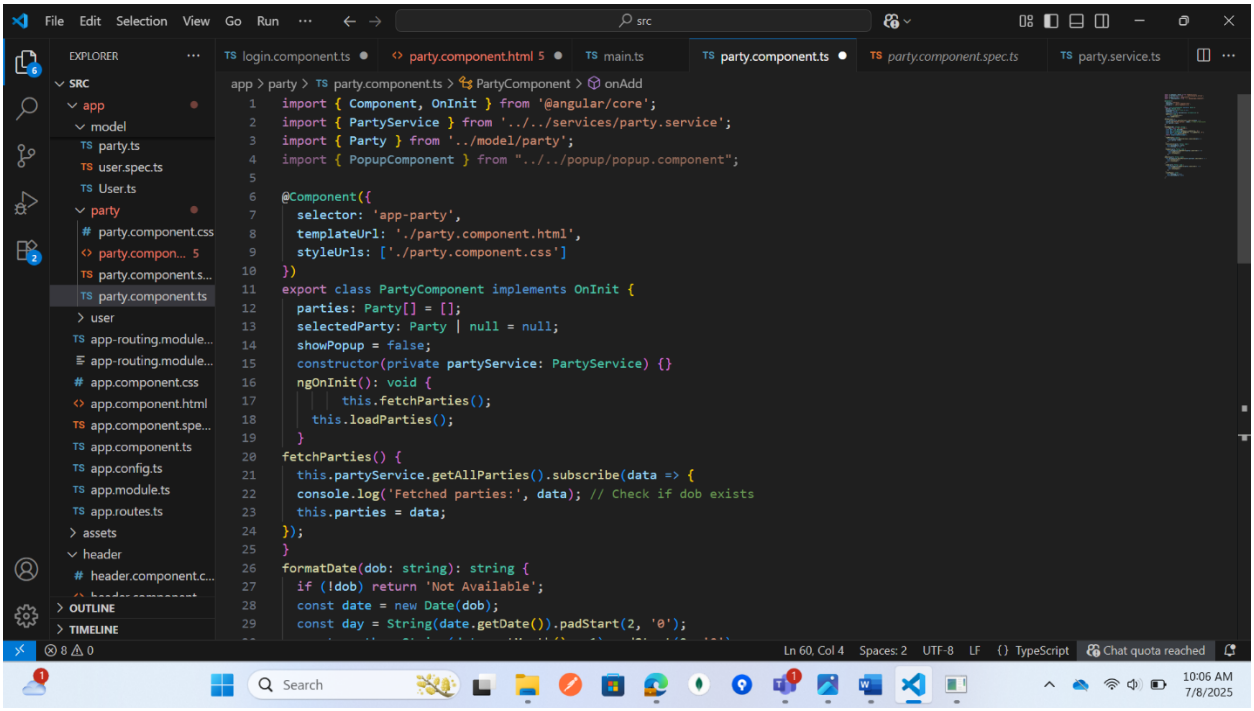




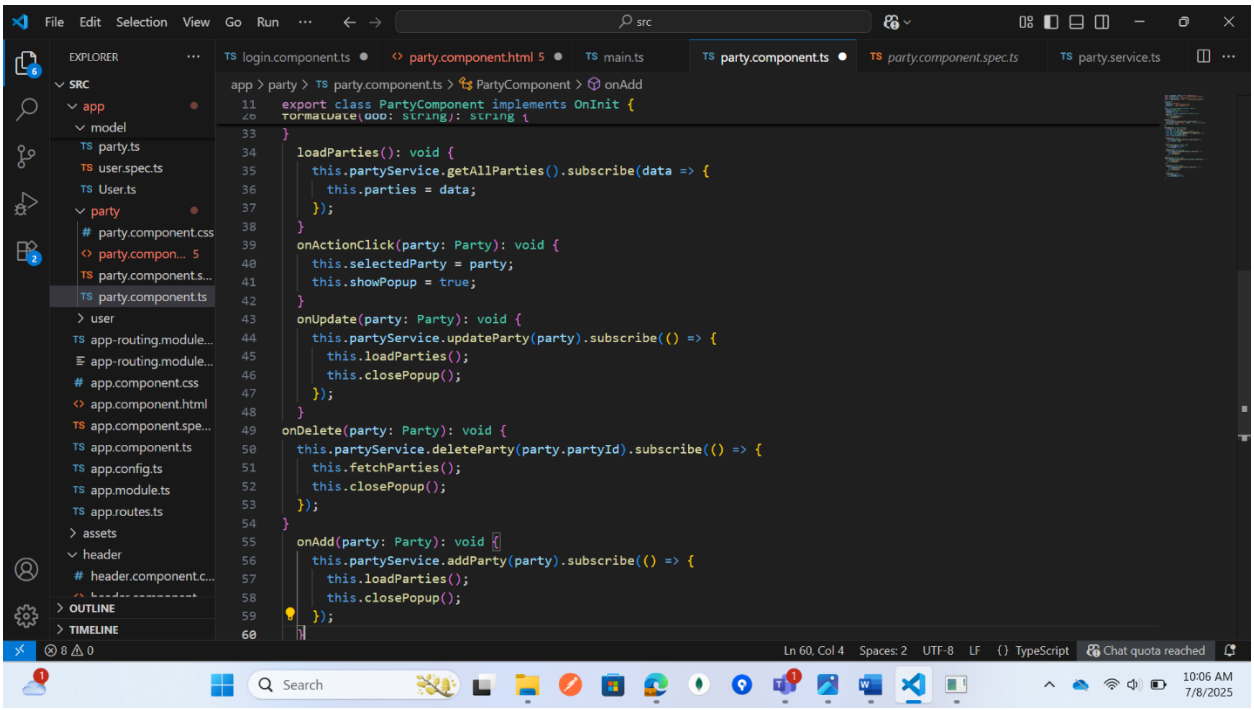
Part.component.html



Part.component.ts

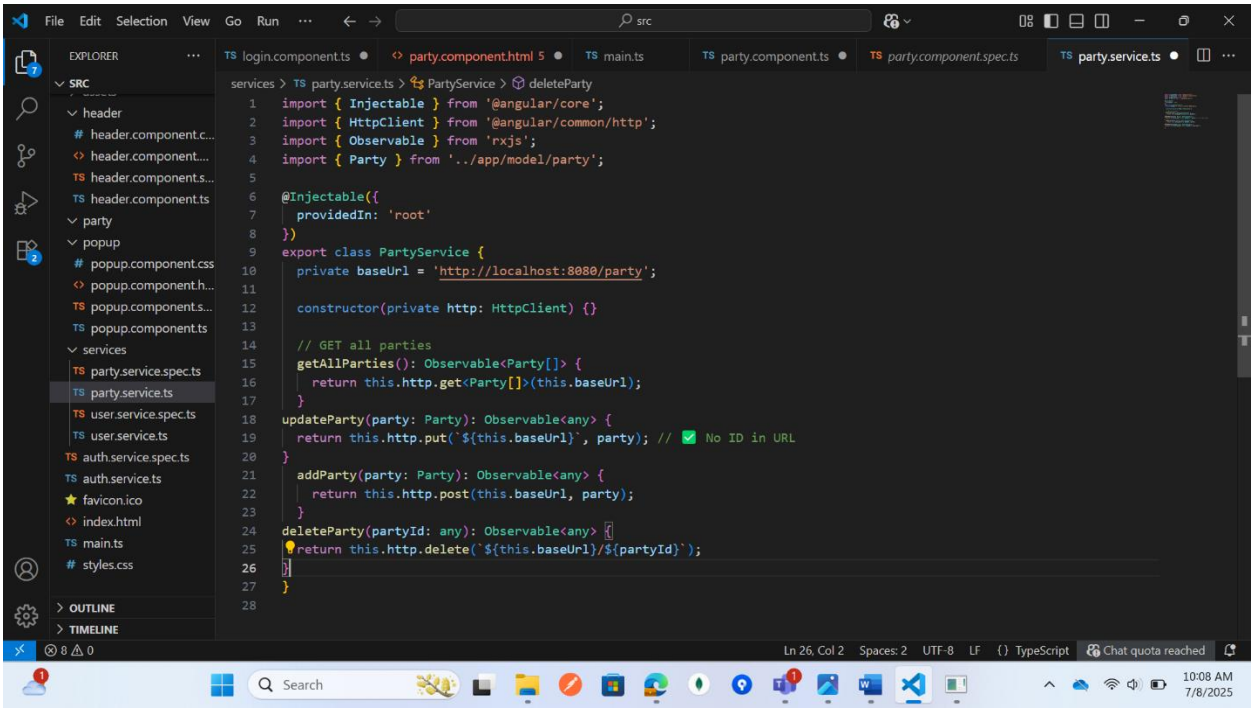


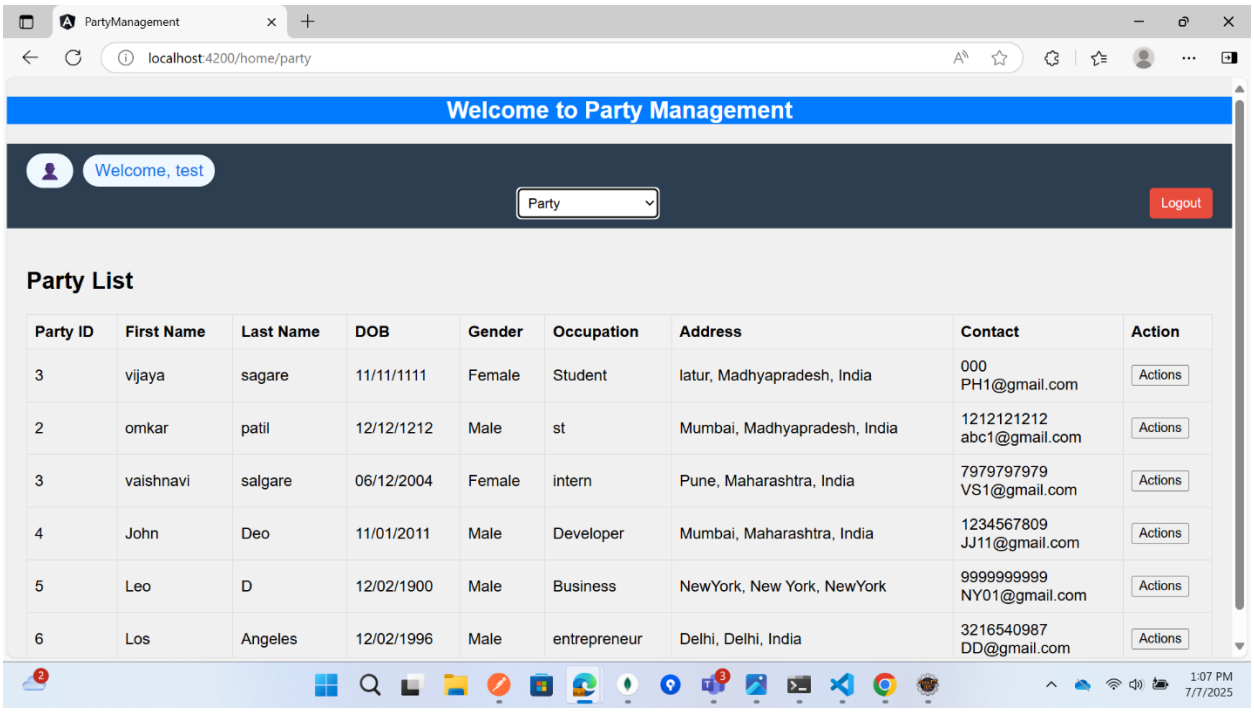
```
import { Component, OnInit } from '@angular/core';
import { PartyService } from '../../services/party.service';
import { Party } from '../model/party';
import { PopupComponent } from "../../popup/popup.component";

@Component({
  selector: 'app-party',
  templateUrl: './party.component.html',
  styleUrls: ['./party.component.css']
})
export class PartyComponent implements OnInit {
  parties: Party[] = [];
  selectedParty: Party | null = null;
  showPopup = false;
  constructor(private partyService: PartyService) {}
  ngOnInit(): void {
      this.fetchParties();
    this.loadParties();
  }
fetchParties() {
  this.partyService.getAllParties().subscribe(data => {
  console.log('Fetched parties:', data); // Check if dob exists
  this.parties = data;
});
}
formatDate(dob: string): string {
  if (!dob) return 'Not Available';
  const date = new Date(dob);
  const day = String(date.getDate()).padStart(2, '0');
```



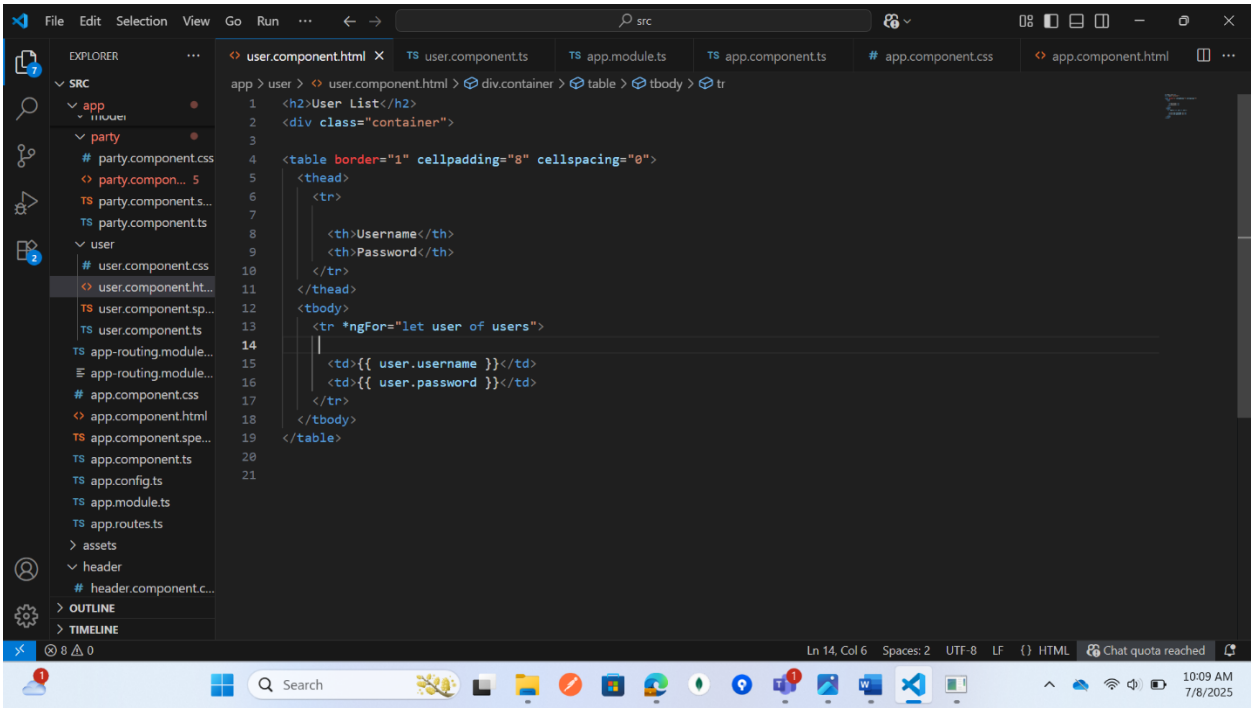
```
export class PartyComponent implements OnInit {
formatDate(dob: string): string {
}
  loadParties(): void {
    this.partyService.getAllParties().subscribe(data => {
      this.parties = data;
    });
  }
  onActionClick(party: Party): void {
    this.selectedParty = party;
    this.showPopup = true;
  }
  onUpdate(party: Party): void {
    this.partyService.updateParty(party).subscribe(() => {
      this.loadParties();
      this.closePopup();
    });
  }
onDelete(party: Party): void {
  this.partyService.deleteParty(party.partyId).subscribe(() => {
    this.fetchParties();
    this.closePopup();
  });
}
  onAdd(party: Party): void {
    this.partyService.addParty(party).subscribe(() => {
      this.loadParties();
      this.closePopup();
    });
}
```

Party.service.ts

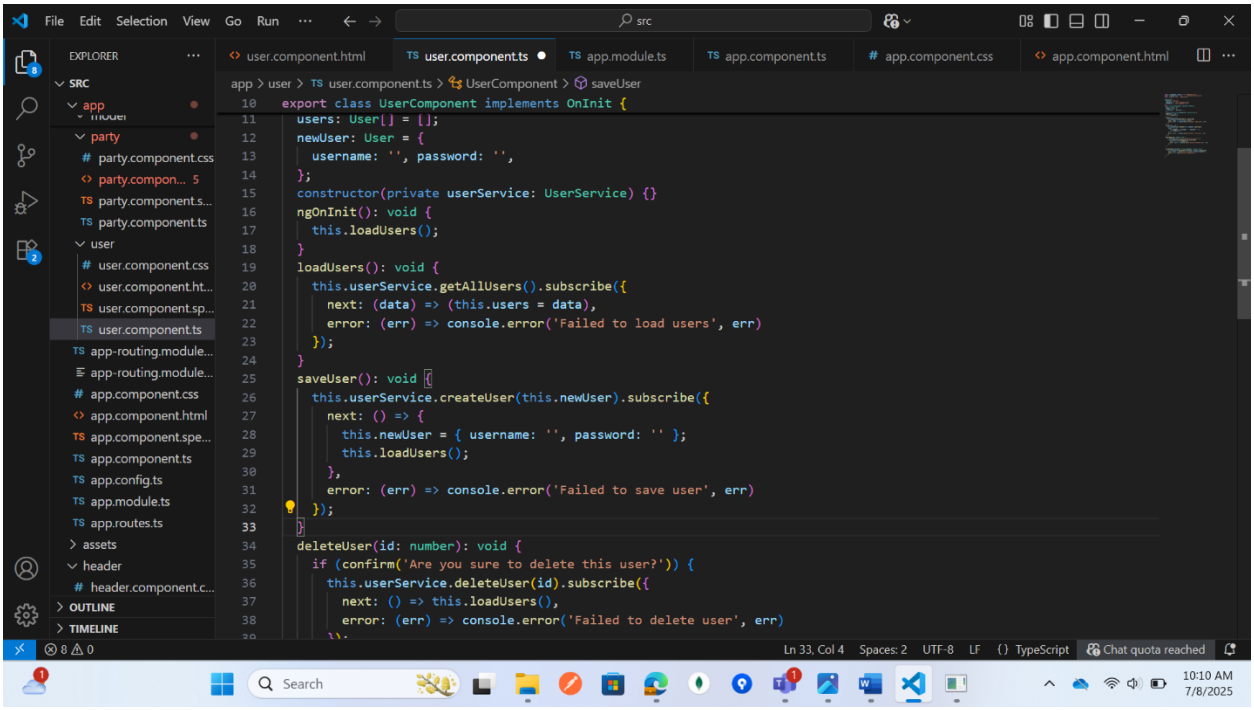




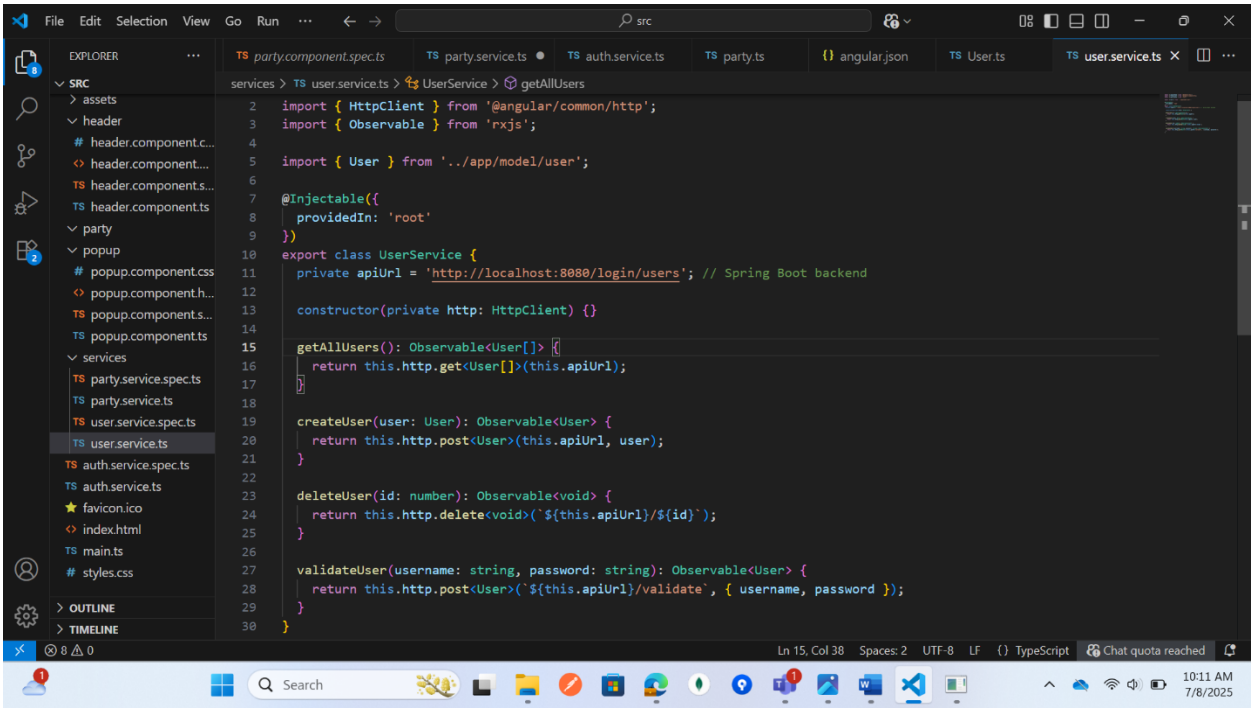
User.component.html

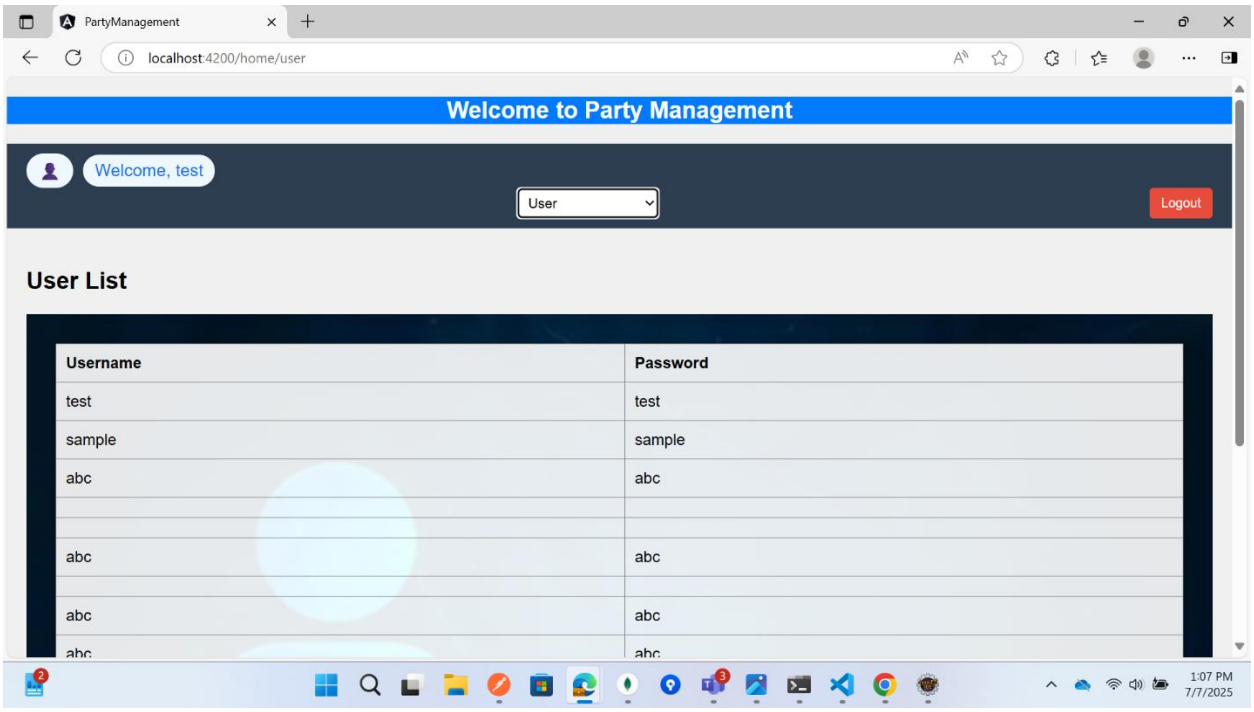


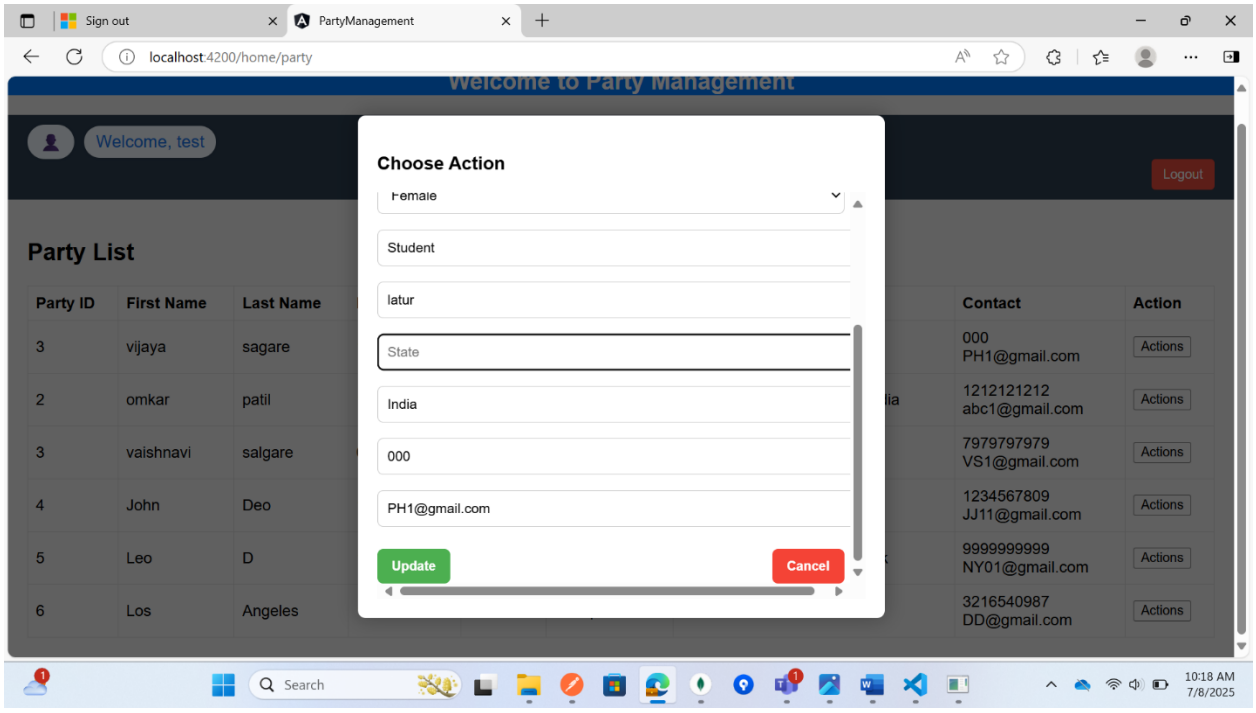
User.component.ts

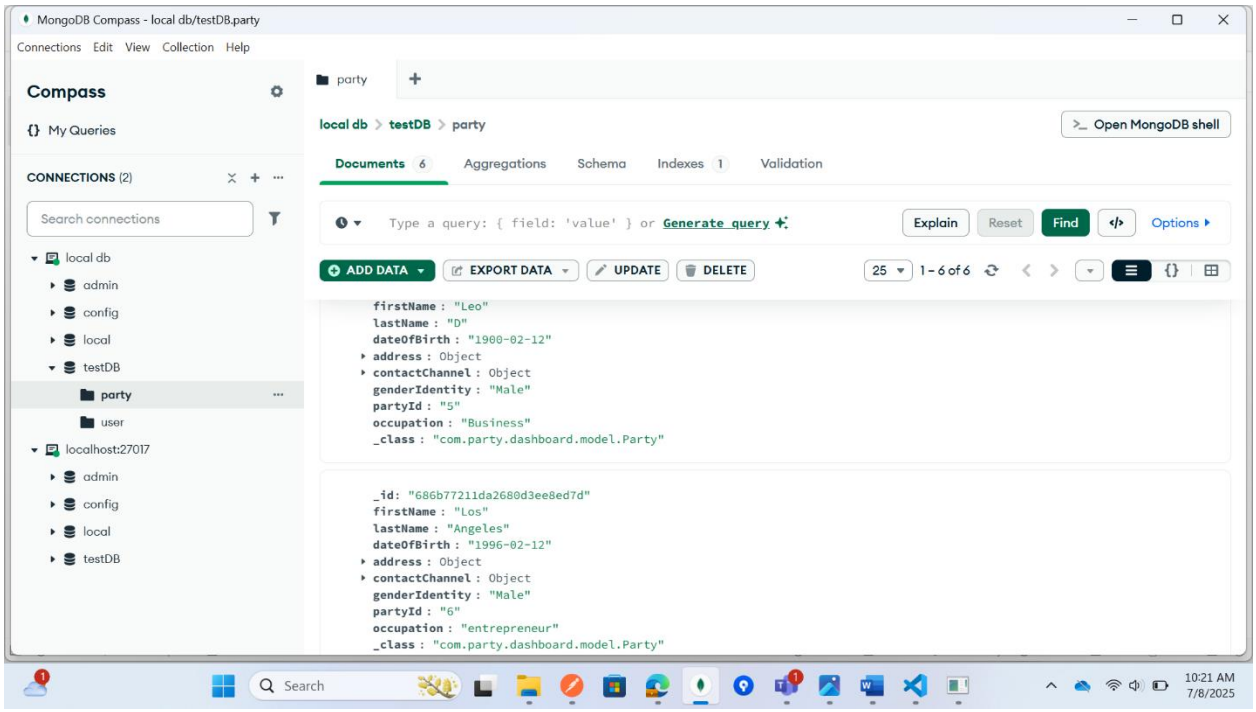


User.service.ts

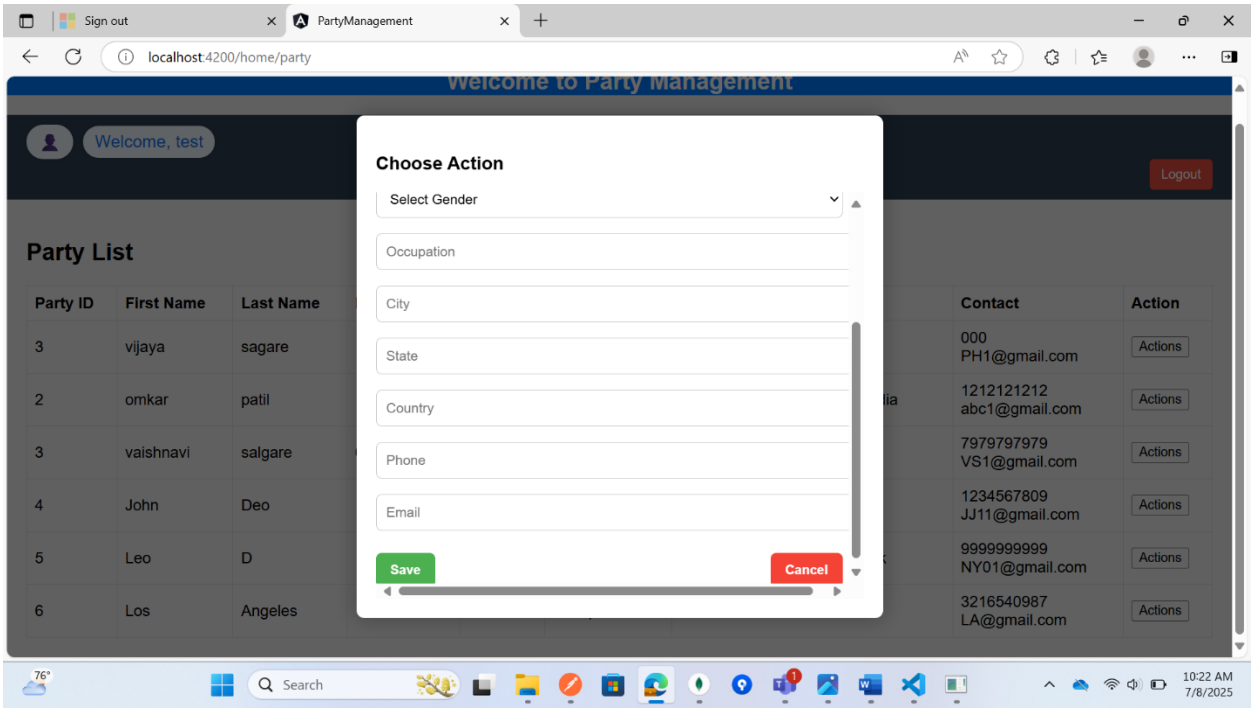

File Edit Selection View Go Run
src
EXPLORER
party.component.spec.ts
party.service.ts
auth.service.ts
party.ts
angular.json
User.ts
user.service.ts
services > user.service.ts > UserService > getAllUsers
SRC
assets
header
party
popup
services
OUTLINE
TIMELINE
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { User } from '../app/model/user';
@Injectable({
providedIn: 'root'
})
export class UserService {
private apiUrl = 'http://localhost:8080/login/users'; // Spring Boot backend
constructor(private http: HttpClient) {}
getAllUsers(): Observable<User[]> {
return this.http.get<User[]>(this.apiUrl);
}
createUser(user: User): Observable<User> {
return this.http.post<User>(this.apiUrl, user);
}
deleteUser(id: number): Observable<void> {
return this.http.delete<void>(`${this.apiUrl}/${id}`);
}
validateUser(username: string, password: string): Observable<User> {
return this.http.post<User>(`${this.apiUrl}/validate`, { username, password });
}
}
Ln 15, Col 38
Spaces: 2
UTF-8
LF
TypeScript
Chat quota reached
Search
10:11 AM
7/8/2025


PartyManagement
localhost:4200/home/user
Welcome to Party Management
Welcome, test
User
Logout
User List
Username
Password
test
test
sample
sample
abc
abc
abc
abc
abc
abc
1:07 PM
7/7/2025





- To Add New Party:



- **Angular Technologies:**
    1. Two-Way Data Binding
    2. Dependency Injection (DI)
    3. Routing & Navigation
    4. Services & HTTP Client
    5. Pipes