A

PROJECT REPORT

ON

Three-C (Connect Contact Conduct)

Submitted in partial fulfillment for the award of

Post Graduate Diploma in Advance Computing

(PG-DAC) from

INSTITUTE OF EMERGING TECHNOLOGIES

Authorized Training Centre

Under the Guidance of

Mrs. Prachi Godbole

BY

Name of student/s and Prn

Nukalwar Parag Sambayya 220945920054

Markad Akash Balkrishna 220945920047

Attar Aijaj Peermohammad 220945920004

Kadam Babalu Dnyandev 220945920038

CERTIFICATE

This is to certify that the project report entitled **Three-C(Connect Contact Conduct)** is a bonfire work carried out by **Nukalwar Parag Sambaya, Markad Akash Balkrishna, Attar Aijaj Peermohammad, Kadam Babalu Dnyandev** and submitted in partial fulfilment of the requirement for the C-DAC ACTS, DAC course in Institute of Emerging Technology in the batch of Aug 2019.

Course Coordinator

Salil Bagade

External Examiner

ACKNOWLEDGEMENT

This project **Three-C(Connect Contact Conduct)** was agreat learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC).

We are very glad to mention the **Mrs. Prachi Godbole** for her valuable guidance to work on this project. Her guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

Our most heart full thanks goes to ***Mr. Sangram Patil*** **(Director ,IET)**who gave all the required support and kind coordination to provide all the necessities like required hardware , internet facility and extra lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune.

Sign of student

Name of Student/s

Roll no.

Nukalwar Parag Sambayya 20945920054

Markad Akash Balkrishna 220945920047

Attar Aijaj Peermohammad 20945920004

Kadam Babalu Dnyandev 220945920038

Abstract

3C(Connect, Contact, Conduct) is a software to connect “Consumers” and “Service provider” with each other to fulfill the need of the Consumer. It is Consumer flexible software which will allow consumer to post about the problem and the necessary service provider will bid on the post, Consumer will be able to select one bid and the Service Provider of that particular bid will get the job. This system will help the freelancers to grow their business and showcase their skills. From consumer’s point of view it will solve the regular household problems which are important but not feasible for consumer due to lack of required skills. It will also reduce the hassle of searching for services and service provider.

Index

1 Introduction

3C(Connect, Contact, Conduct) is a software to connect “Consumers” and “Service provider” with each other to fulfill the need of the Consumer. It is Consumer flexible software which will allow consumer to post about the problem and the necessary service provider will bid on the post, Consumer will be able to select one bid and the Service Provider of that particular bid will get the job. This system will help the freelancers to grow their business and showcase their skills. From consumer’s point of view it will solve the regular household problems which are important but not feasible for consumer due to lack of required skills. It will also reduce the hassle of searching for services and service provider.

2 Problem definition and Scope

2.1 Problem Definition

In the fast life of city it is difficult to get the service needed to solve the daily needs of the consumer like plumbing, maid, electric issues etc , especially for the people who are new to the city. Three C caters the gap between Consumer and Service Provider to give a desired service to the consumer based on Post-Bid service.

2.2 Goals and Objectives

3C(Connect, Contact, Conduct) is a software to connect “Consumers” and “Service provider” with each other. It is a platform which will cater to the everyday problems of consumers - Plumbing, Maids, Electrical Repairs etc - which can be outsourced to the professionals. Also, it will create opportunities for freelancers to grow their businesses and skills. The platform will solve everyday household problems which are important but not feasible for consumer due to lack of required skills. Consumer will post about the problem. The Service providers will bid on it and Consumer will select the most convenient amongst all.

3 Software Requirement Specification

The objective of SRS document is to formally describe the system's high level requirement including functional requirement, non-functional requirement.

3.1 Proposed System

The System Three C will cater the bridge between Consumer and Service Provider and the system is designed around Consumer which means Consumer will make a post on the problem and based on the category the Service Provider will make a bid, so it is designed to be a Consumer friendly system.

3.2 Scope

Consumer will have the freedom of selecting a time slot according to his convenience. Service Provider will be able to see all posts related to his/her skills on 3C platform. He can bid on the most economic work-post according to his convenience. It will build trust between the engaging parties. It will be more economic for the parties to carry their business through trusted 3C platform. The platform will secure economic interest of parties without compromising work quality and security.

4 System Modules

Consumer using this software should get registered and provide his details like Name, Phone number and email ID to the system.

Service provider who is willing to sell service should get registered and provide his details like Name, Address, Phone number, Qualification details, Skills and email ID to the system.

Registered Consumer will be able to make a post about his problems(Title, description) based on a particular skills required to solve the problem. Consumer will be able to add date according to his/her convenience and Consumer must specify his locality/Area.

Registered Service provider will be able to see all posts based on his/her skills and would be able to bid on the post but he/she will not be able to see other bids.

When Consumer receives all the bids he will select ONE bid amongst all and the notification of acceptance will be sent to the service provider, after selecting the

bid Consumer can now share all the details (Name, Phone, Postal Address etc) to the service provider.

Once the bid is accepted by Consumer, service provider must do it and if the service provider is willing to cancel it then service provider must pay nominal fine to the system and inform to the system.

5 Performance-Requirements

Security
Registered Consumer will be allowed to post on the platform to seek service.
Each stakeholder will be able to access the system through authentication process.
System will provide access to the content , operations using Role based security (Authorization) (Permissions based on Role)
Using SSL in all transactions which will be performed stakeholder. It would protect confidential information shared by system to
System will automatically log of all stakeholder after some time due to inactiveness.
System will block operations for inactive stakeholder and would redirect for authentication.
System will internally maintain secure communication channel between Servers (Web Servers, App Servers, database Server)
Sensitive data will be always encrypted across communication.
Use proper firewall to protect servers from outside phishing, vulnerable attacks.

Reliability
The system will backup business data on regular basis and recover in short time duration to keep system operational
Continuous updates are maintained , continuous Administration is done to keep system operational.
During peak hours system will maintain same user experience by managing load balancing .

Availability
uptime: 24* 7 available
99.999%

Maintainability:
A Commercial database software will be used to maintain System data Persistence.
A readymade Web Server will be installed to host online site Web Site to management server capabilities.

IT operations team will easily monitor and configure System using Administrative tools provided by Servers.
Separate environment will be maintained for system for isolation in production, testing, and development.

Portability:
PDA: Portable Device Application
System will provide portable User Interface (ReactJS, CSS, JSX) through users will be able to access online freelancing platform.
System can be deployed to single server, multi-server, to any OS, Cloud (AWS)

Accessibility:
Only the registered Users will be able to post or bid on the platform. Admin will be able to see all the posts and bids.

Durability:

System will implement backup and recovery for retaining stake holders data, business operation data and business data over time.

Efficiency:
System will be able to manage all transactions with isolation.

Modularity:
System will be designed and developed using reusable, independent or dependent business scenarios in the form of modules.
These modules will be loosely coupled and highly cohesive.
System will contain the history of the posts, details of bids received, status of the work, payment status.
System will contain the work history of the service provider, personal details , ratings and review details.
modules.

Scalability:
System will be able to provide consistent user experience to stake holders as well as visitors irrespective of load.

Safety:

System will be secure from malicious attack, phishing.
System functionalities are protected from outside with proper firewall configuration.

System will be always kept updated with latest antivirus software.
Business data will be backed up periodically to ensure safety of data using incremental back up strategy.

5.1 H/W Requirements & S/W Requirements

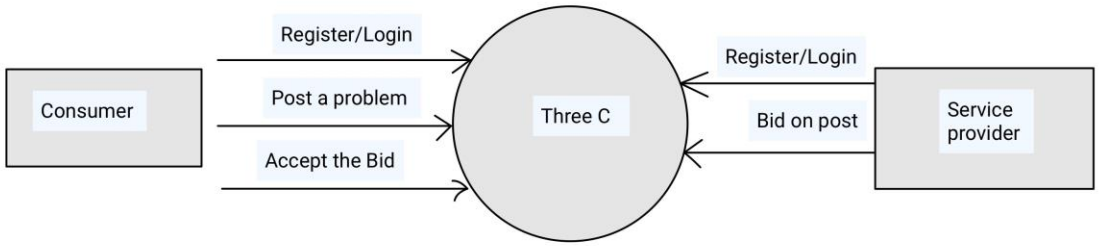
Java (SpringBoot)
Javascript (ReactJS)
Eclipse IDE
Visual Studio Code

6 UML Diagram

6.1 DFD (Data Flow Diagram)

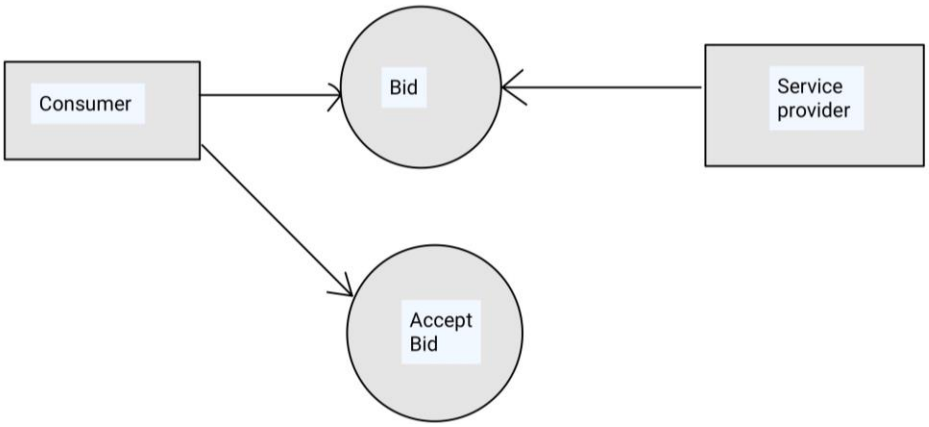
0-level DFD:

It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows



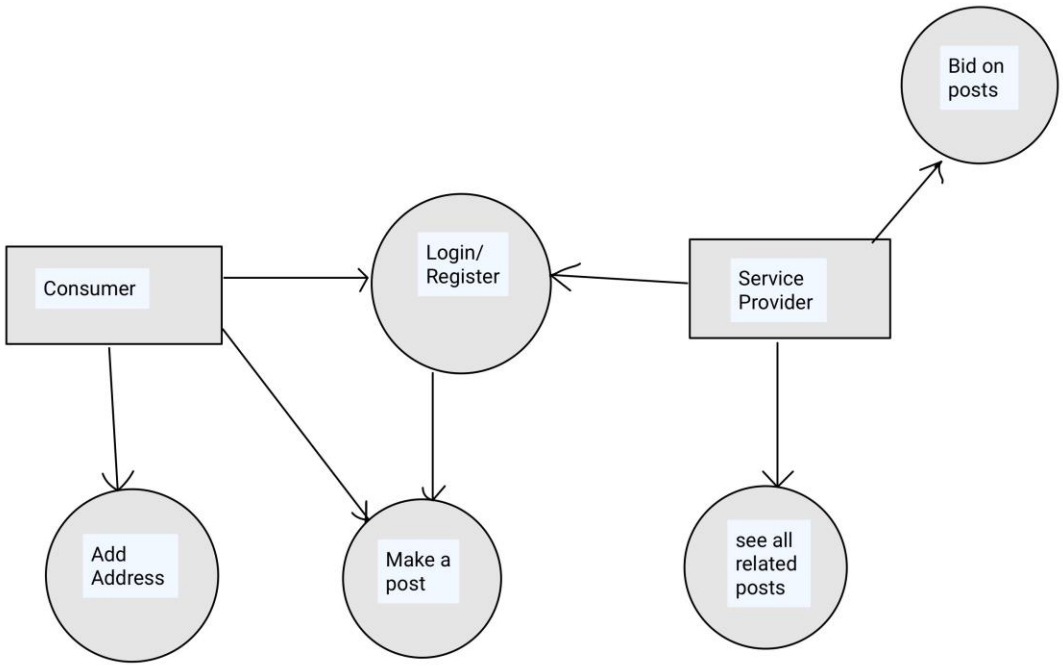
1-level DFD:

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.



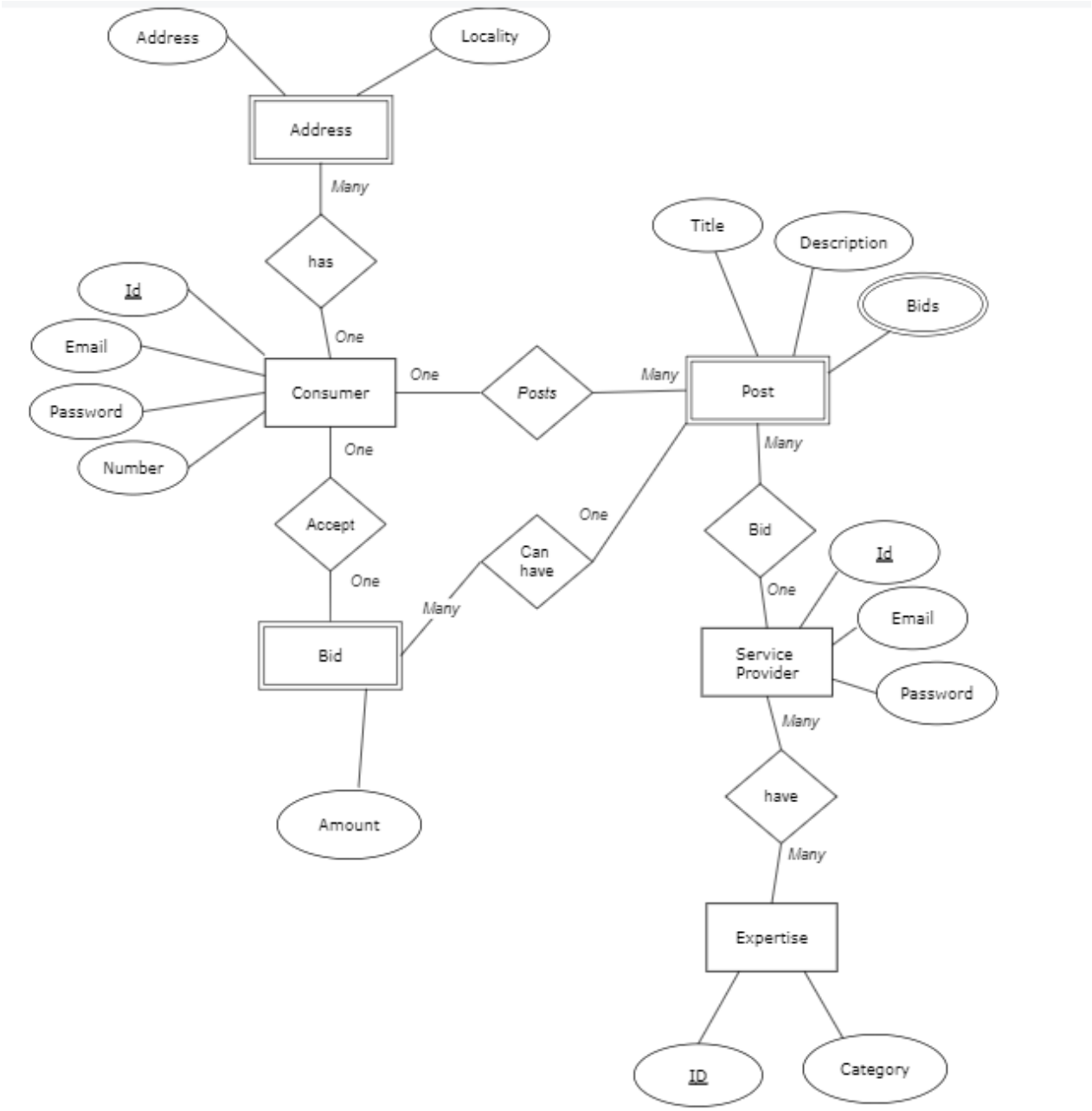
2-level DFD:

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.



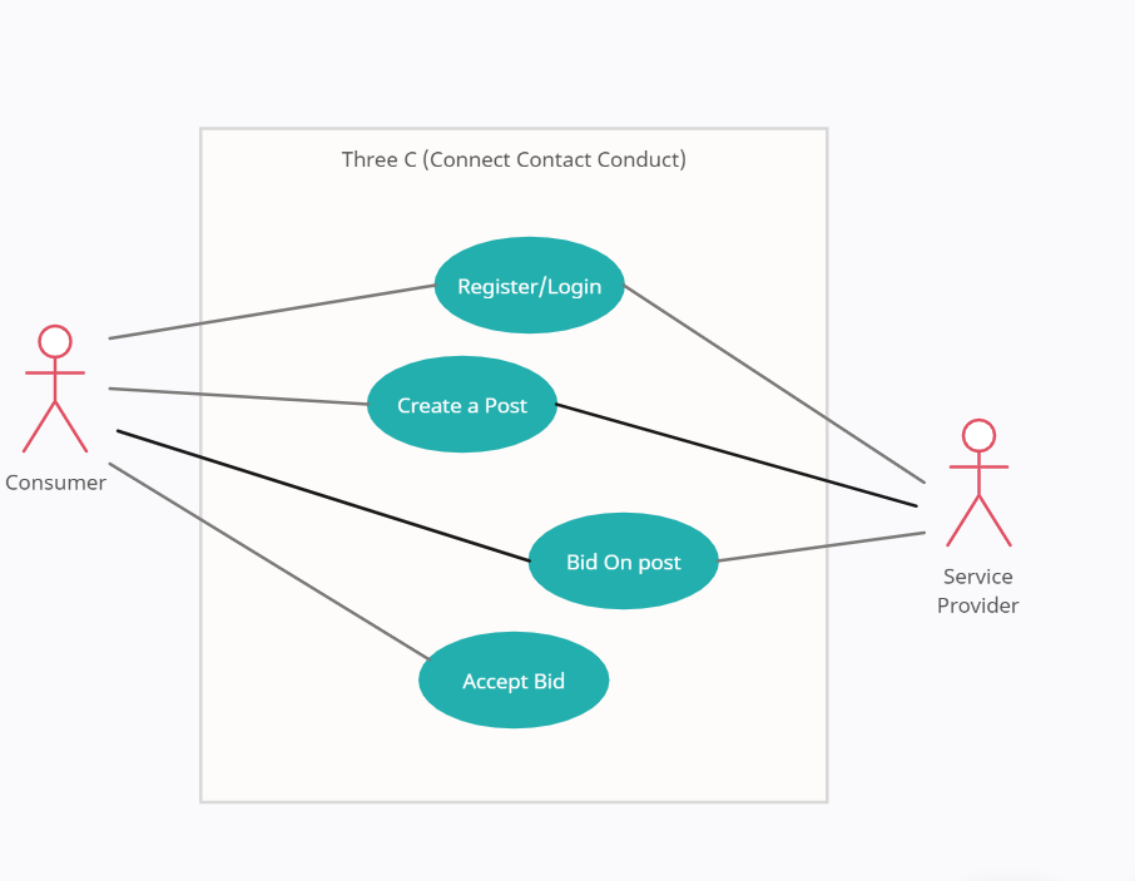
6.2 ERD (Entity Relationship Diagram)

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system.



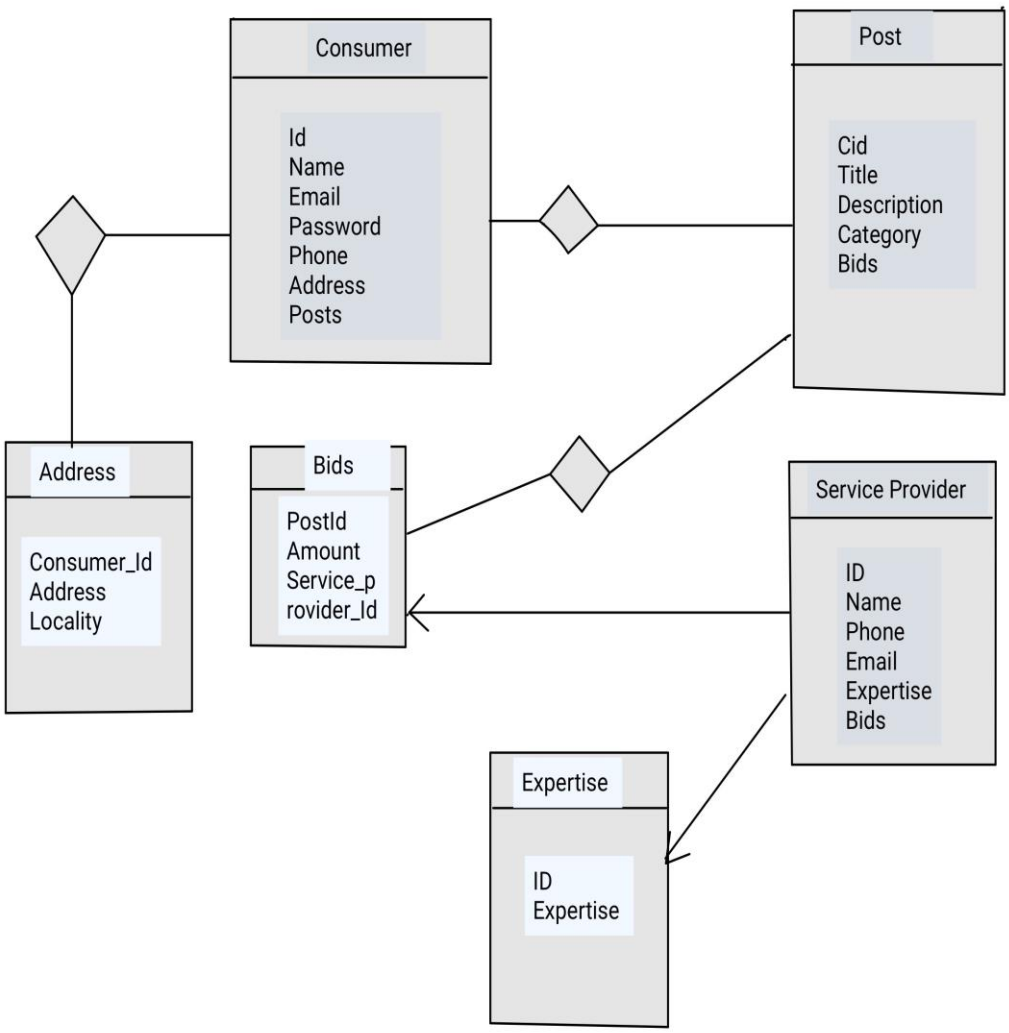
6.3 Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors.



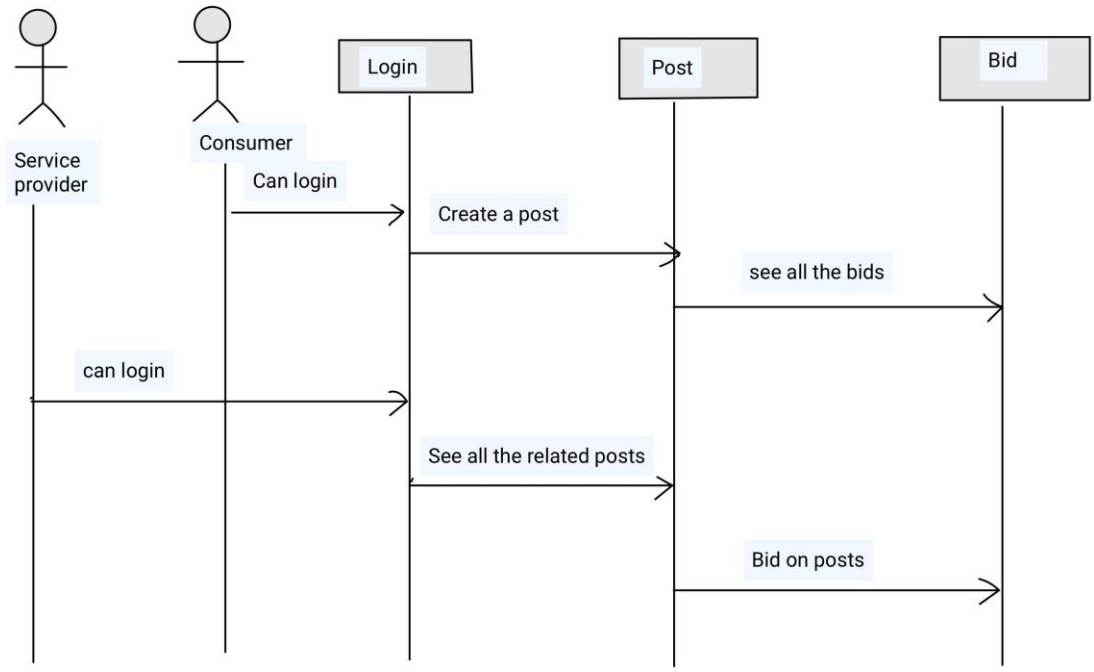
6.4 Class Diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.



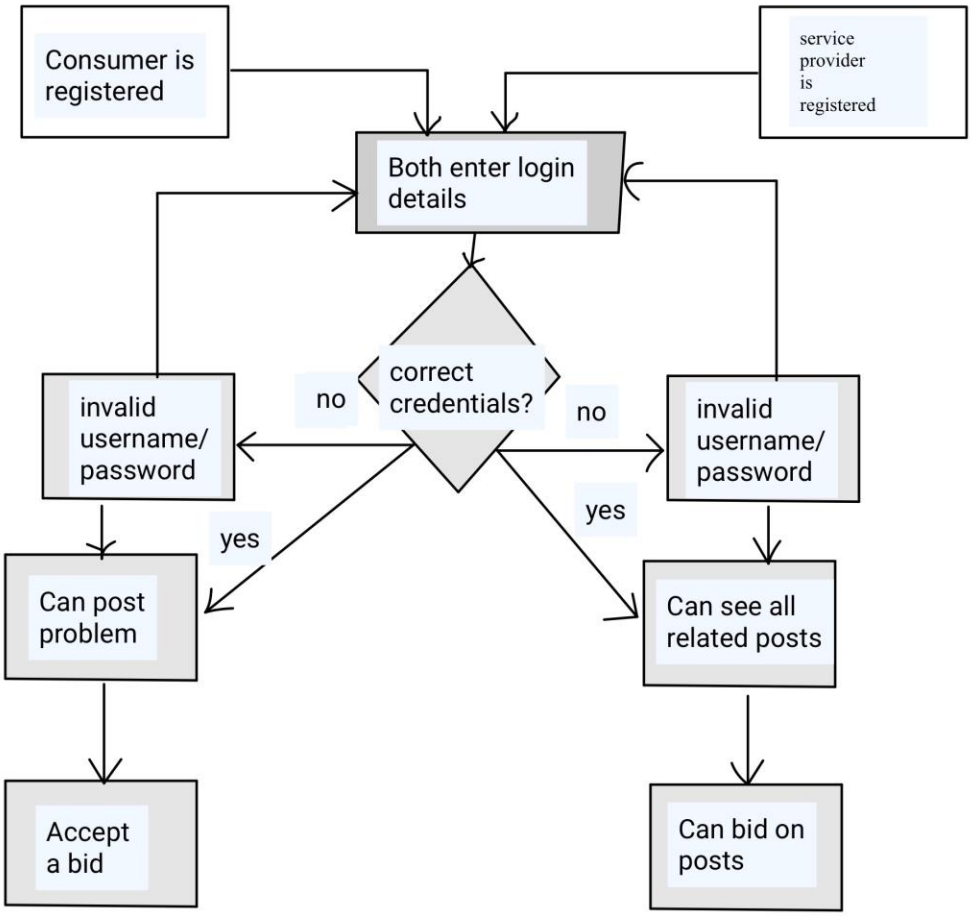
6.5 Sequence Diagram

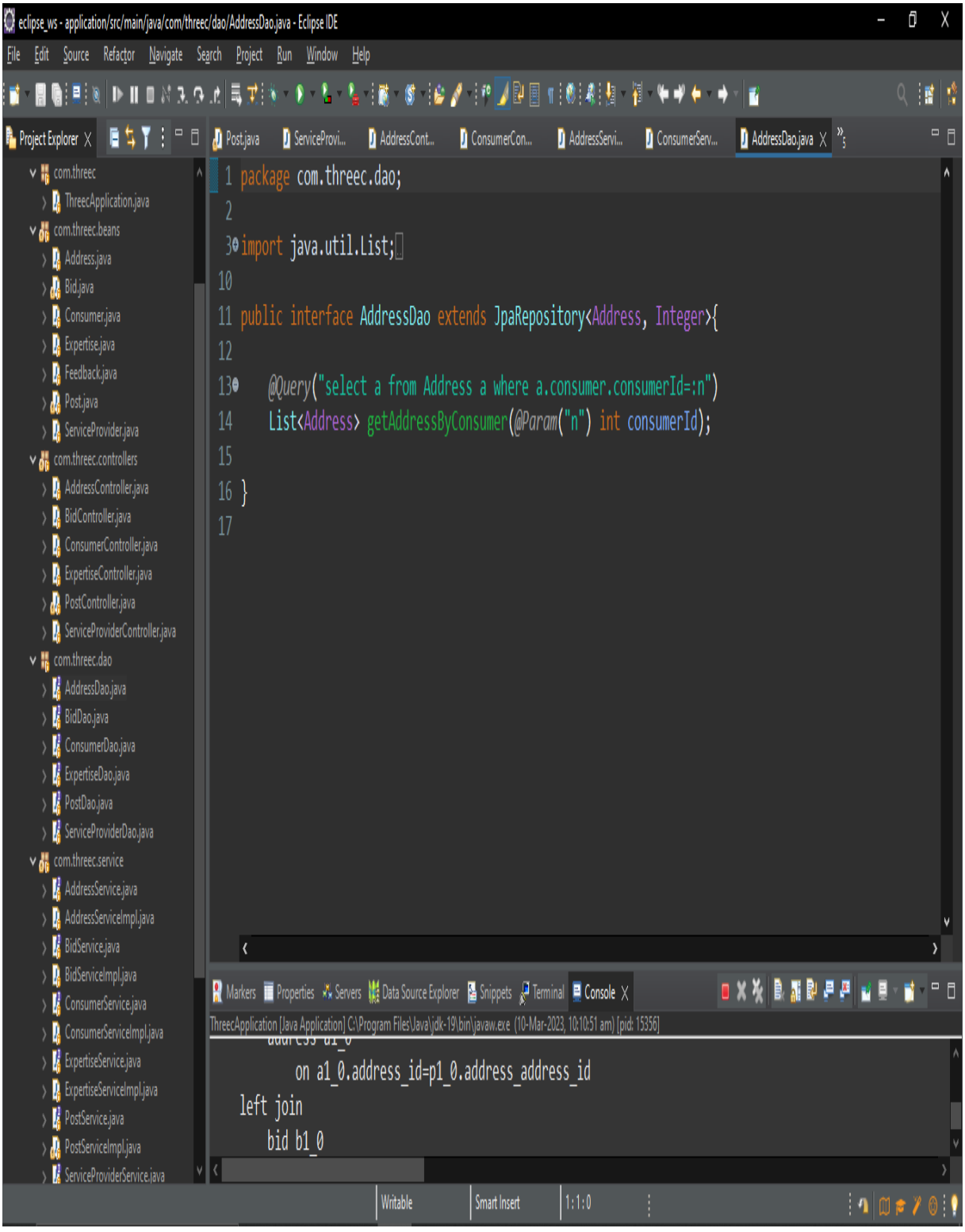
A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

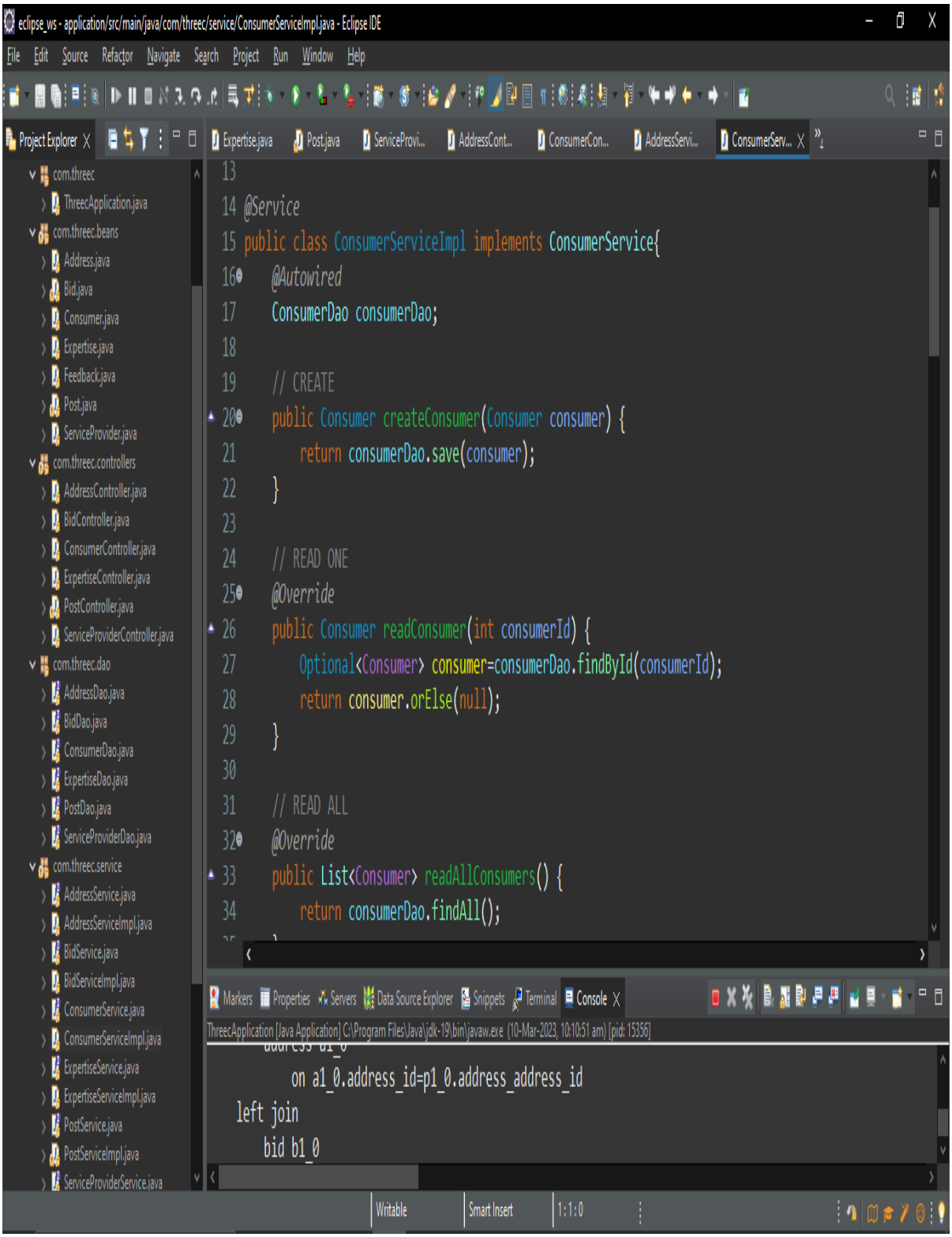


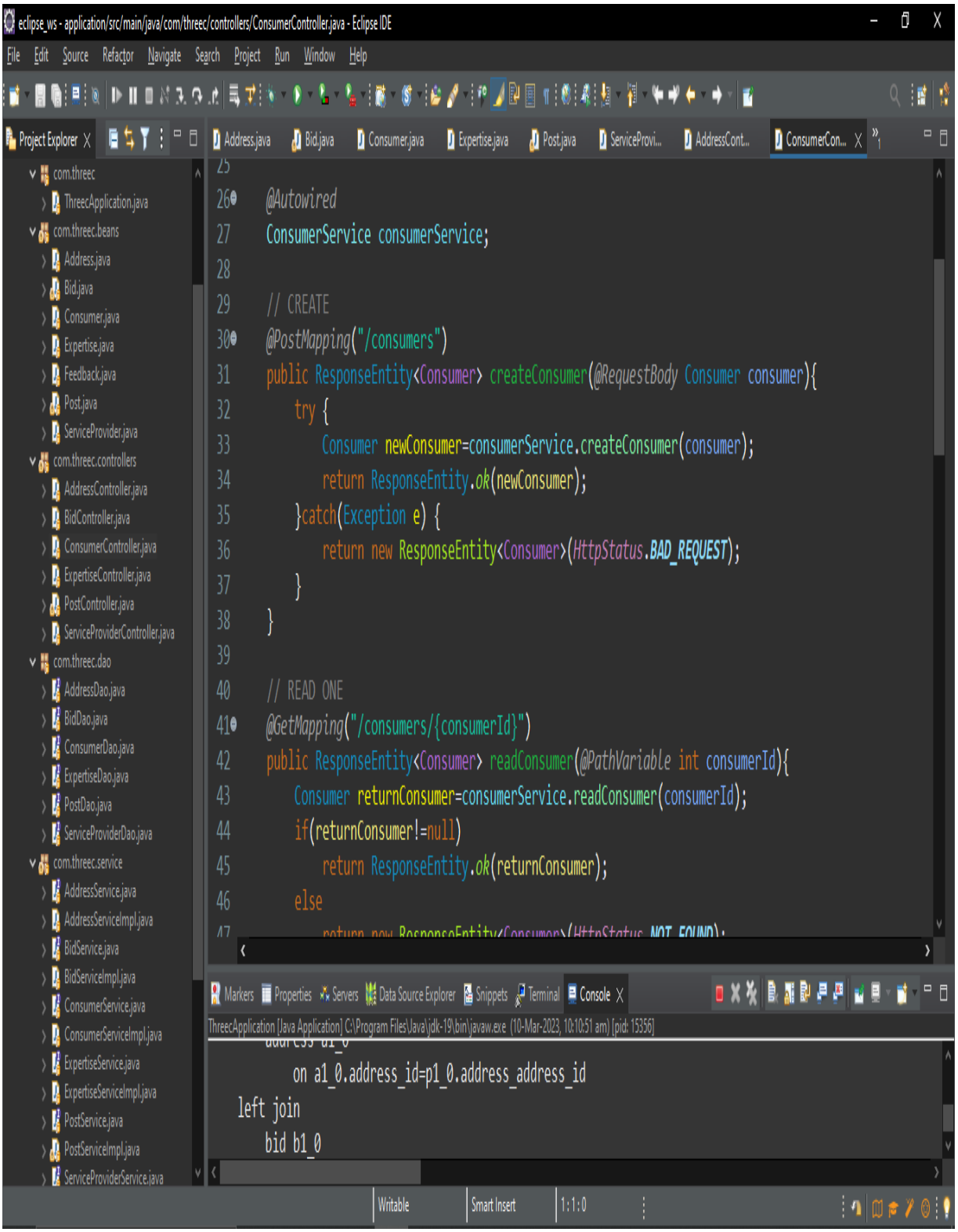
6.6 Activity Diagram

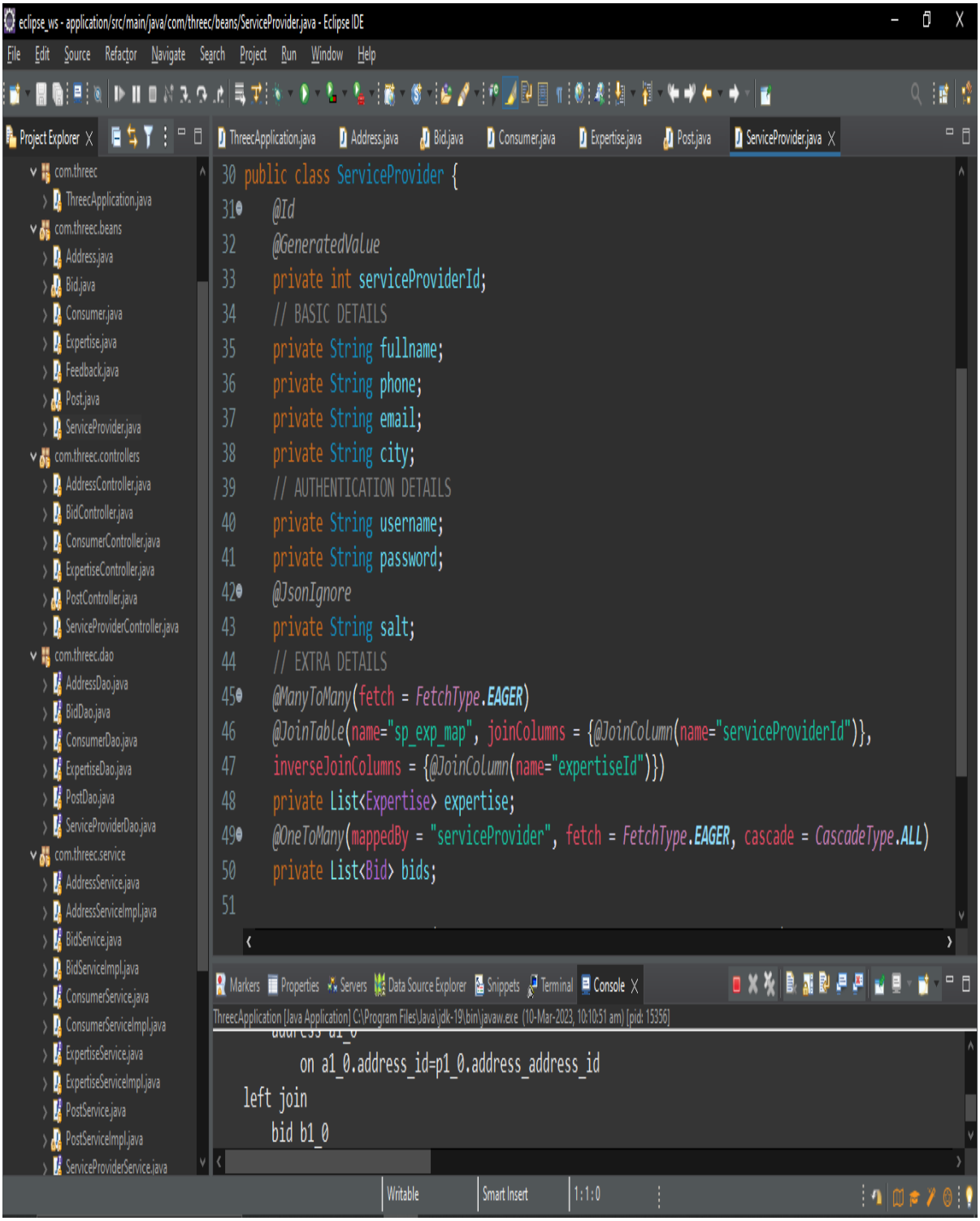
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

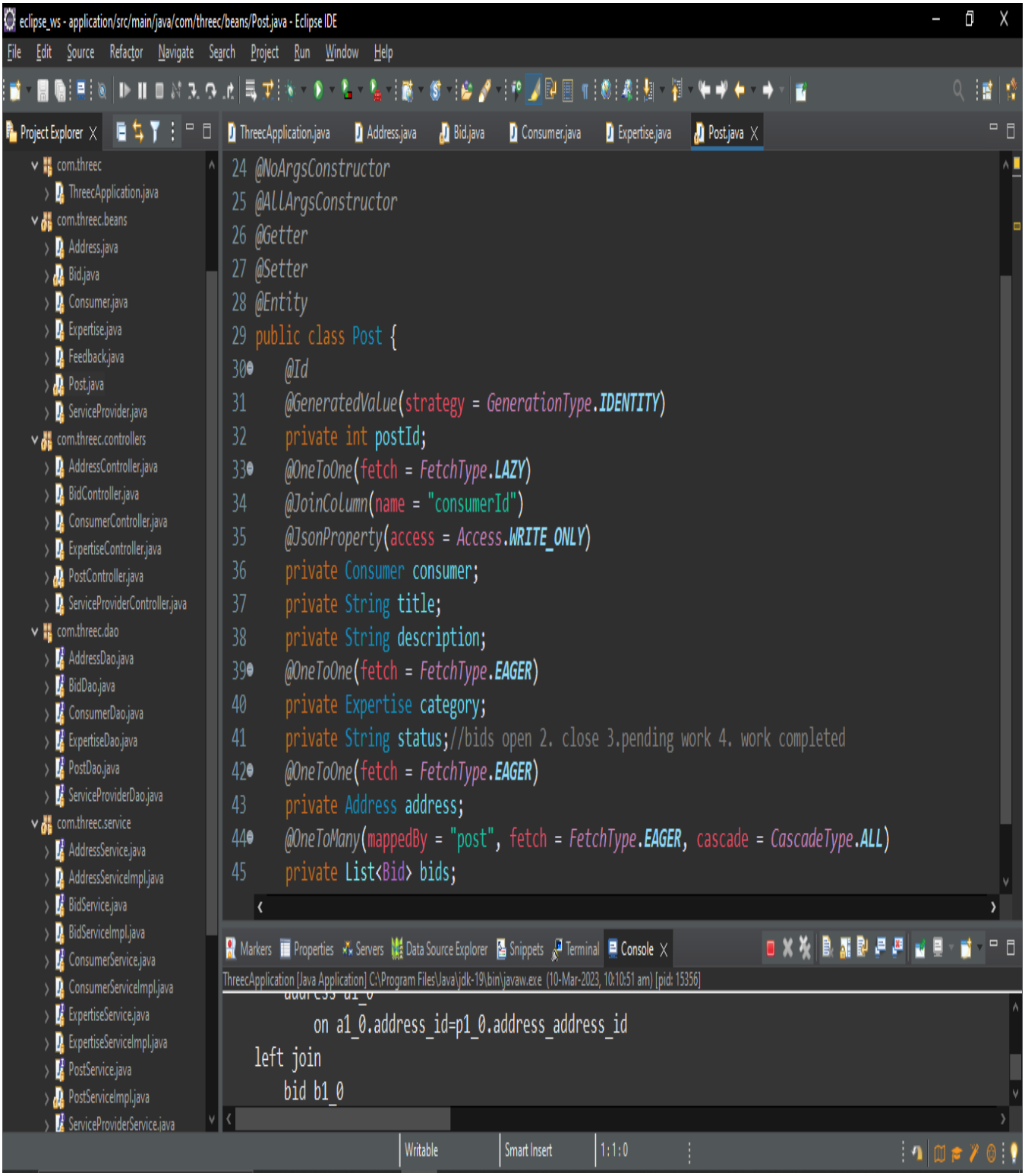


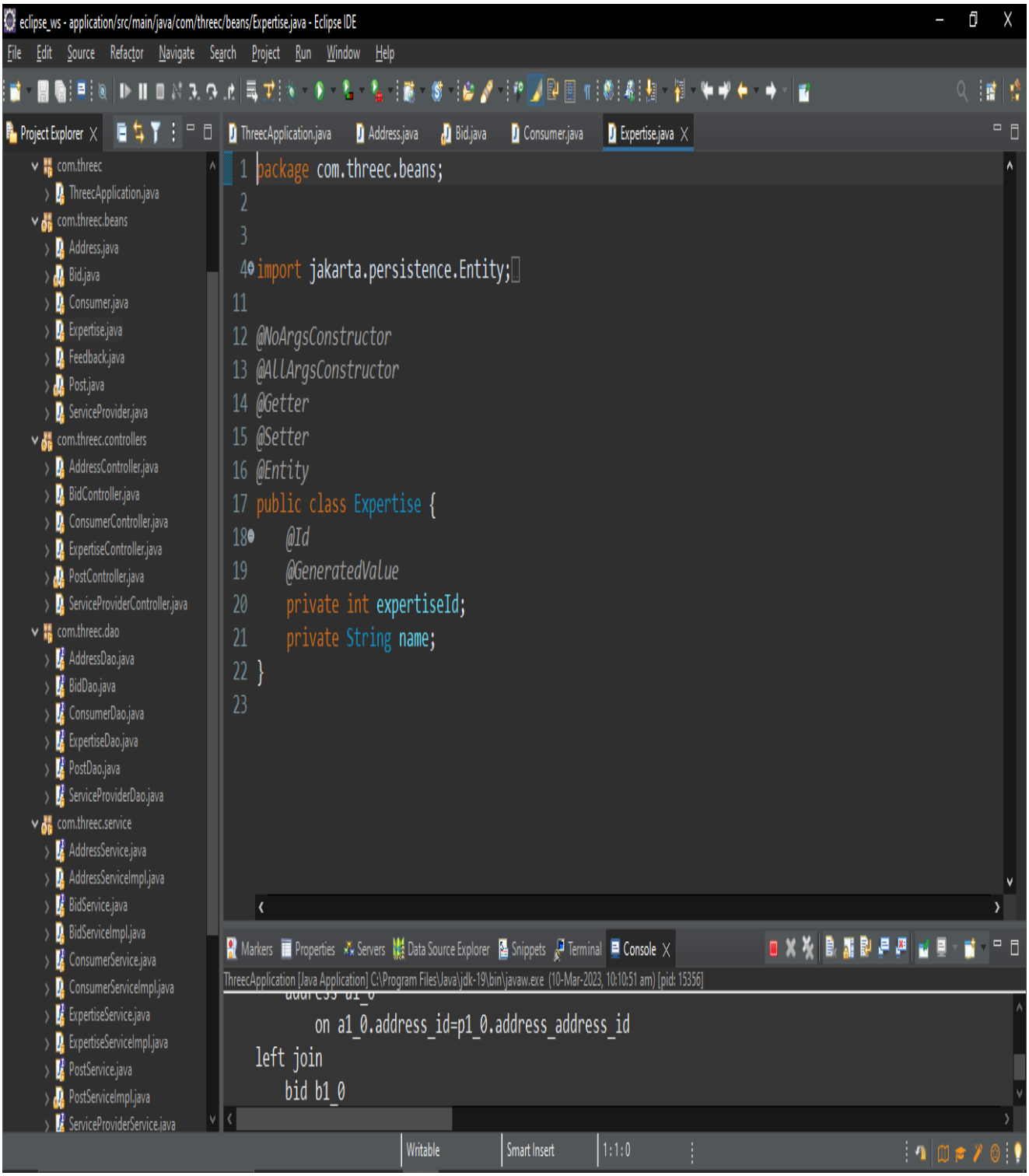

eclipse_ws - application/src/main/java/com/threec/dao/AddressDao.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
Post.java
ServiceProvi...
AddressCont...
ConsumerCon...
AddressServi...
ConsumerServ...
AddressDao.java
1 package com.threec.dao;
2
3 import java.util.List;
10
11 public interface AddressDao extends JpaRepository<Address, Integer>{
12
13 @Query("select a from Address a where a.consumer.consumerId=:n")
14 List<Address> getAddressByConsumer(@Param("n") int consumerId);
15
16 }
17
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

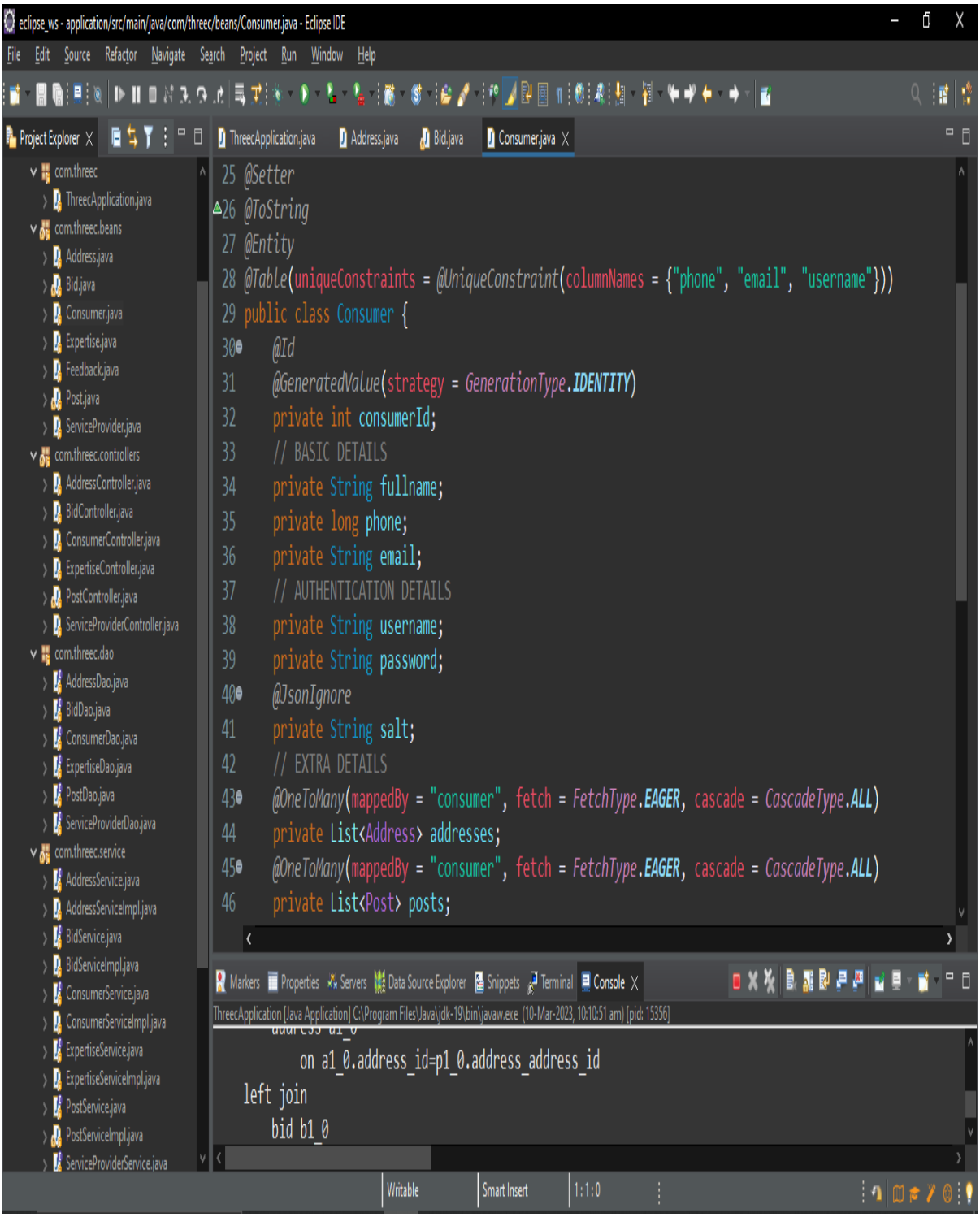

eclipse_ws - application/src/main/java/com/threec/service/ConsumerServiceImpl.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
Expertise.java
Post.java
ServiceProvi...
AddressCont...
ConsumerCon...
AddressServi...
ConsumerServ...
13
14 @Service
15 public class ConsumerServiceImpl implements ConsumerService{
16 @Autowired
17 ConsumerDao consumerDao;
18
19 // CREATE
20 public Consumer createConsumer(Consumer consumer) {
21 return consumerDao.save(consumer);
22 }
23
24 // READ ONE
25 @Override
26 public Consumer readConsumer(int consumerId) {
27 Optional<Consumer> consumer=consumerDao.findById(consumerId);
28 return consumer.orElse(null);
29 }
30
31 // READ ALL
32 @Override
33 public List<Consumer> readAllConsumers() {
34 return consumerDao.findAll();
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

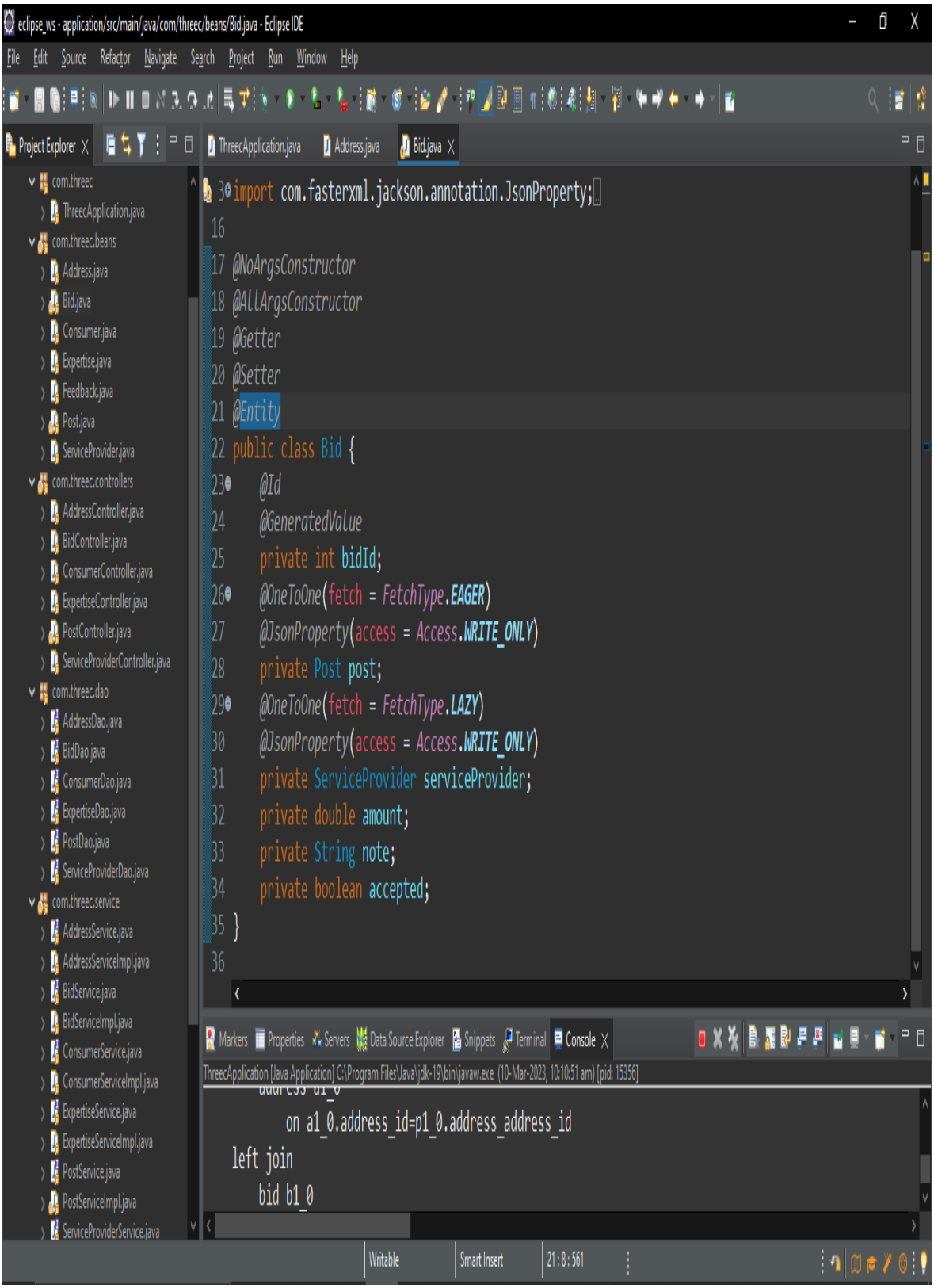

eclipse_ws - application/src/main/java/com/threec/controllers/ConsumerController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
Address.java
Bid.java
Consumer.java
Expertise.java
Post.java
ServiceProvi...
AddressCont...
ConsumerCon...
@Autowired
ConsumerService consumerService;
// CREATE
@PostMapping("/consumers")
public ResponseEntity<Consumer> createConsumer(@RequestBody Consumer consumer){
try {
Consumer newConsumer=consumerService.createConsumer(consumer);
return ResponseEntity.ok(newConsumer);
}catch(Exception e) {
return new ResponseEntity<Consumer>(HttpStatus.BAD_REQUEST);
}
}
// READ ONE
@GetMapping("/consumers/{consumerId}")
public ResponseEntity<Consumer> readConsumer(@PathVariable int consumerId){
Consumer returnConsumer=consumerService.readConsumer(consumerId);
if(returnConsumer!=null)
return ResponseEntity.ok(returnConsumer);
else
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

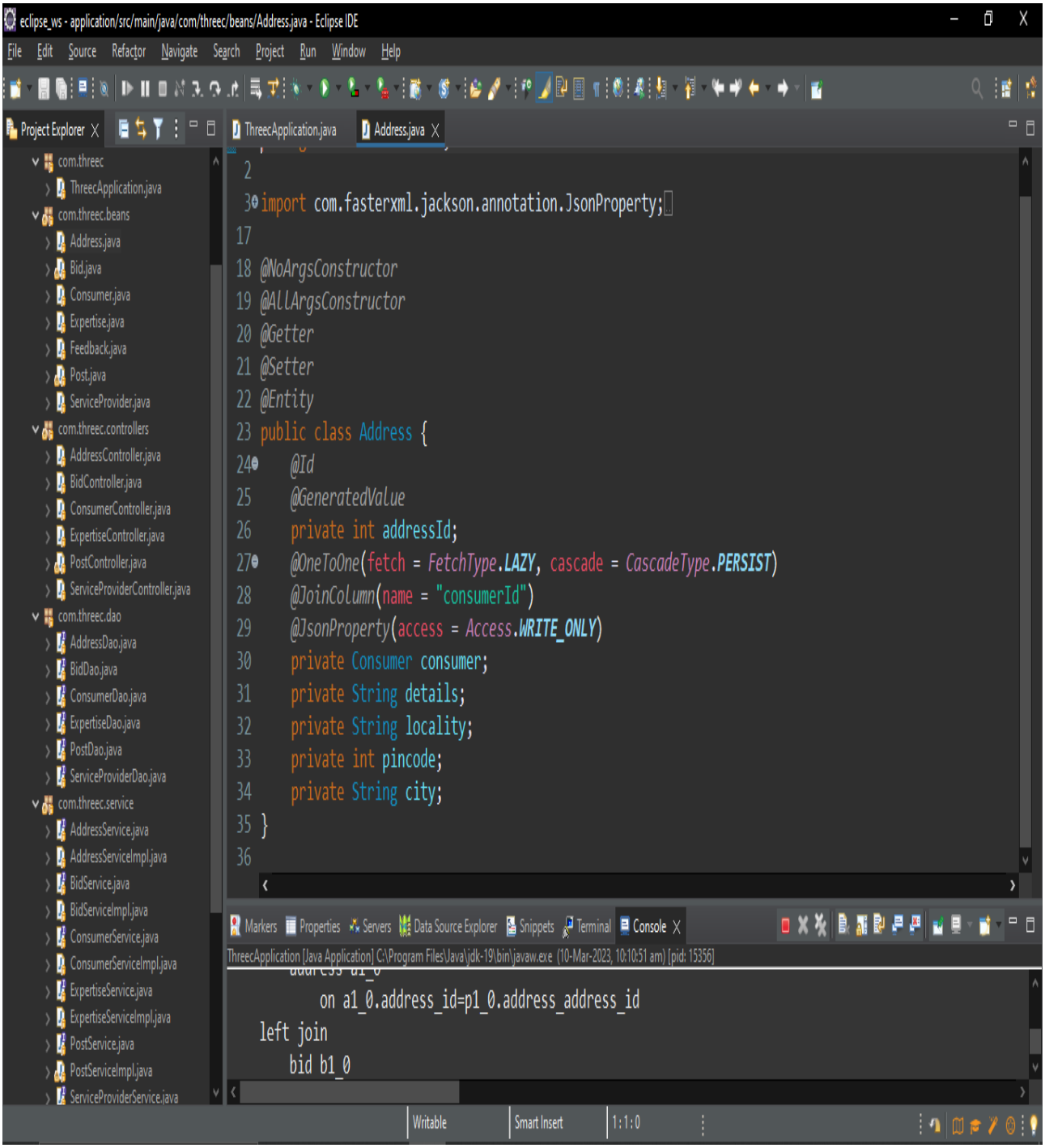

eclipse_ws - application/src/main/java/com/threec/beans/ServiceProvider.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ThreecApplication.java
Address.java
Bid.java
Consumer.java
Expertise.java
Post.java
ServiceProvider.java
30 public class ServiceProvider {
31 @Id
32 @GeneratedValue
33 private int serviceProviderId;
34 // BASIC DETAILS
35 private String fullname;
36 private String phone;
37 private String email;
38 private String city;
39 // AUTHENTICATION DETAILS
40 private String username;
41 private String password;
42 @JsonIgnore
43 private String salt;
44 // EXTRA DETAILS
45 @ManyToMany(fetch = FetchType.EAGER)
46 @JoinTable(name="sp_exp_map", joinColumns = {@JoinColumn(name="serviceProviderId")},
47 inverseJoinColumns = {@JoinColumn(name="expertiseId")})
48 private List<Expertise> expertise;
49 @OneToMany(mappedBy = "serviceProvider", fetch = FetchType.EAGER, cascade = CascadeType.ALL)
50 private List<Bid> bids;
51
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

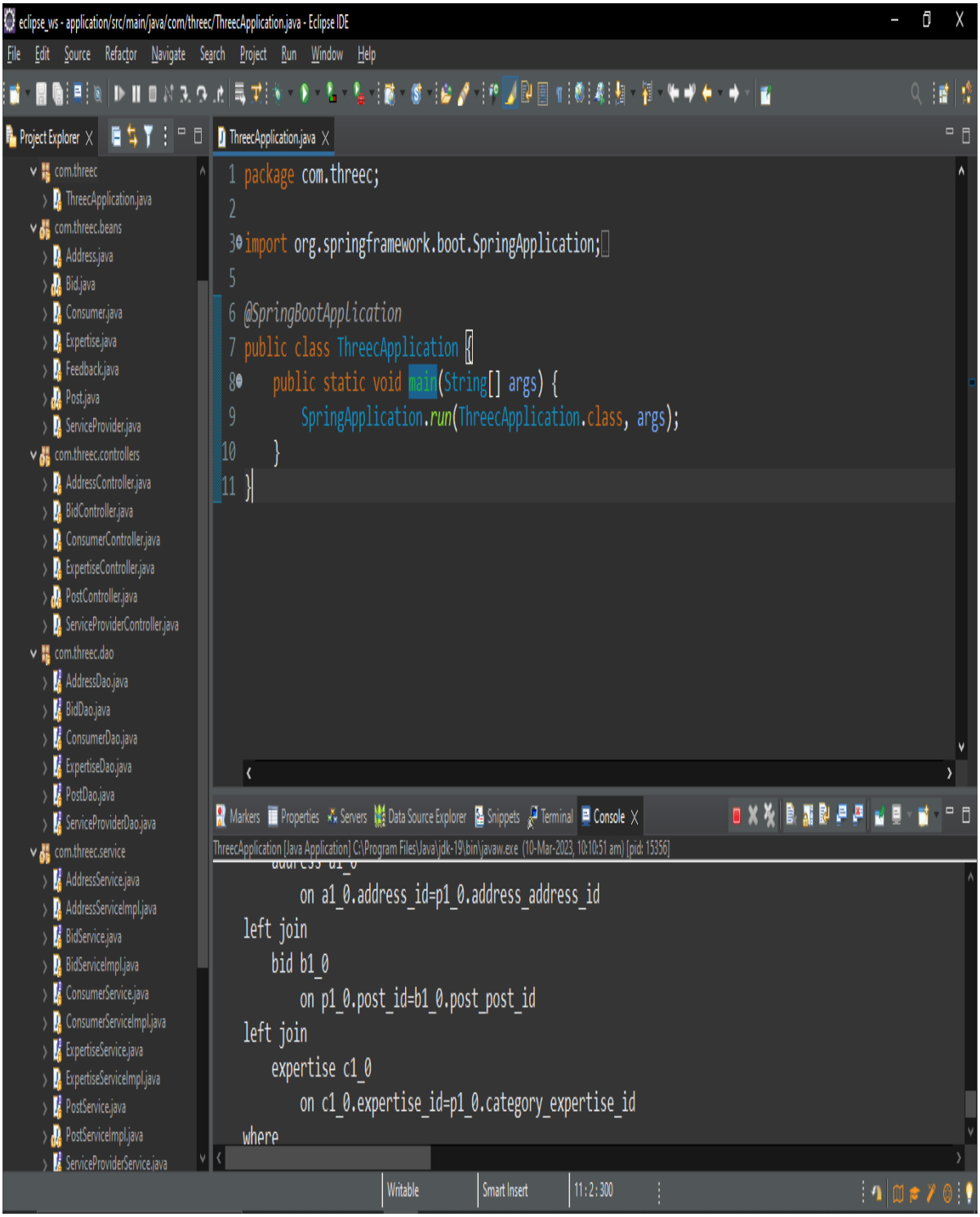

eclipse_ws - application/src/main/java/com/threec/beans/Post.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ThreecApplication.java
Address.java
Bid.java
Consumer.java
Expertise.java
Post.java
24 @NoArgsConstructor
25 @AllArgsConstructor
26 @Getter
27 @Setter
28 @Entity
29 public class Post {
30 @Id
31 @GeneratedValue(strategy = GenerationType.IDENTITY)
32 private int postId;
33 @OneToOne(fetch = FetchType.LAZY)
34 @JoinColumn(name = "consumerId")
35 @JsonProperty(access = Access.WRITE_ONLY)
36 private Consumer consumer;
37 private String title;
38 private String description;
39 @OneToOne(fetch = FetchType.EAGER)
40 private Expertise category;
41 private String status;//bids open 2. close 3.pending work 4. work completed
42 @OneToOne(fetch = FetchType.EAGER)
43 private Address address;
44 @OneToMany(mappedBy = "post", fetch = FetchType.EAGER, cascade = CascadeType.ALL)
45 private List<Bid> bids;
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

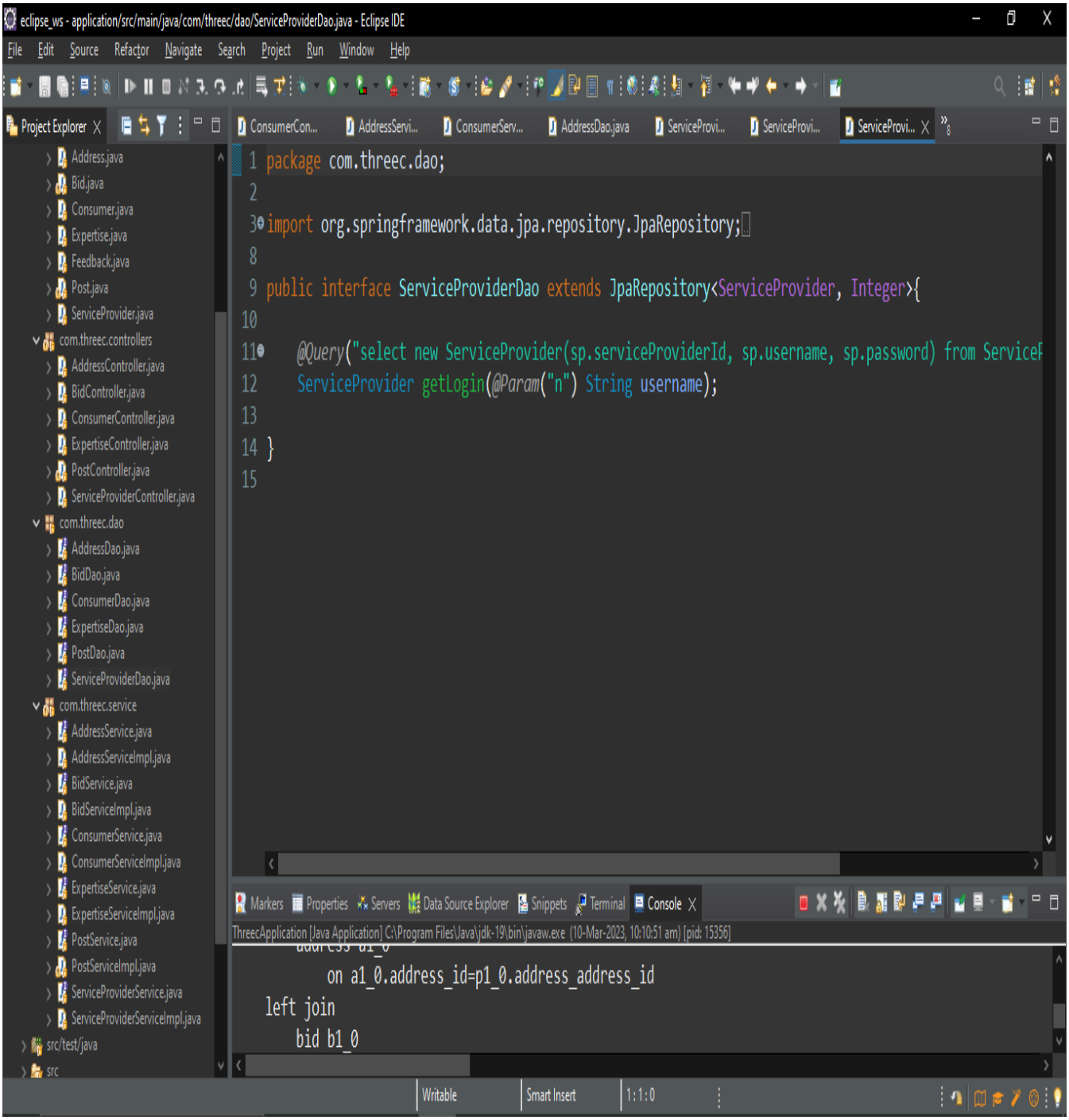

eclipse_ws - application/src/main/java/com/threec/beans/Expertise.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ThreecApplication.java
Address.java
Bid.java
Consumer.java
Expertise.java
1 package com.threec.beans;
2
3
4 import jakarta.persistence.Entity;
11
12 @NoArgsConstructor
13 @AllArgsConstructor
14 @Getter
15 @Setter
16 @Entity
17 public class Expertise {
18 @Id
19 @GeneratedValue
20 private int expertiseId;
21 private String name;
22 }
23
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1 : 1 : 0

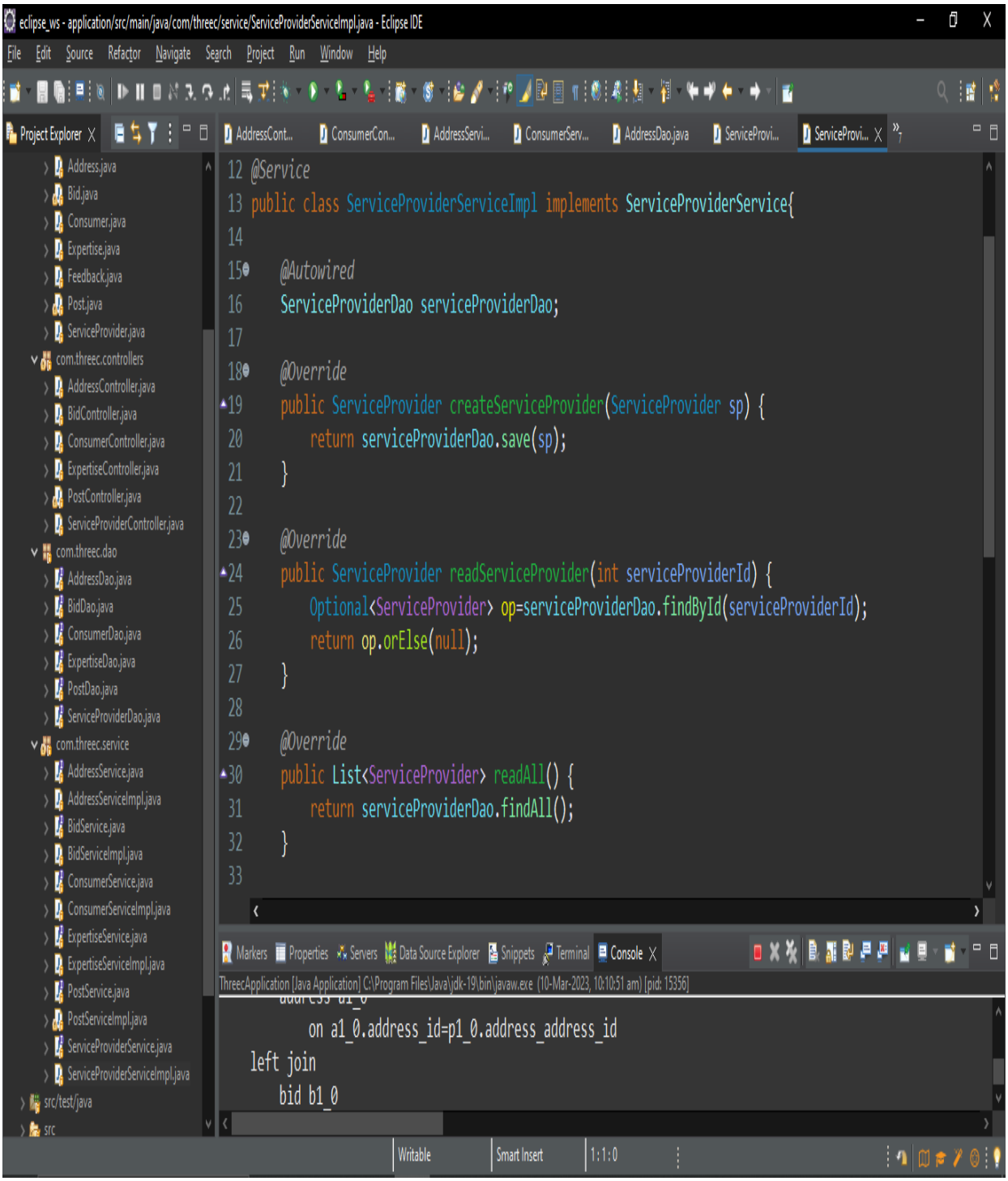

eclipse_ws - application/src/main/java/com/threec/beans/Consumer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ThreecApplication.java
Address.java
Bid.java
Consumer.java
@Setter
@ToString
@Entity
@Table(uniqueConstraints = @UniqueConstraint(columnNames = {"phone", "email", "username"}))
public class Consumer {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int consumerId;
// BASIC DETAILS
private String fullname;
private long phone;
private String email;
// AUTHENTICATION DETAILS
private String username;
private String password;
@JsonIgnore
private String salt;
// EXTRA DETAILS
@OneToMany(mappedBy = "consumer", fetch = FetchType.EAGER, cascade = CascadeType.ALL)
private List<Address> addresses;
@OneToMany(mappedBy = "consumer", fetch = FetchType.EAGER, cascade = CascadeType.ALL)
private List<Post> posts;
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

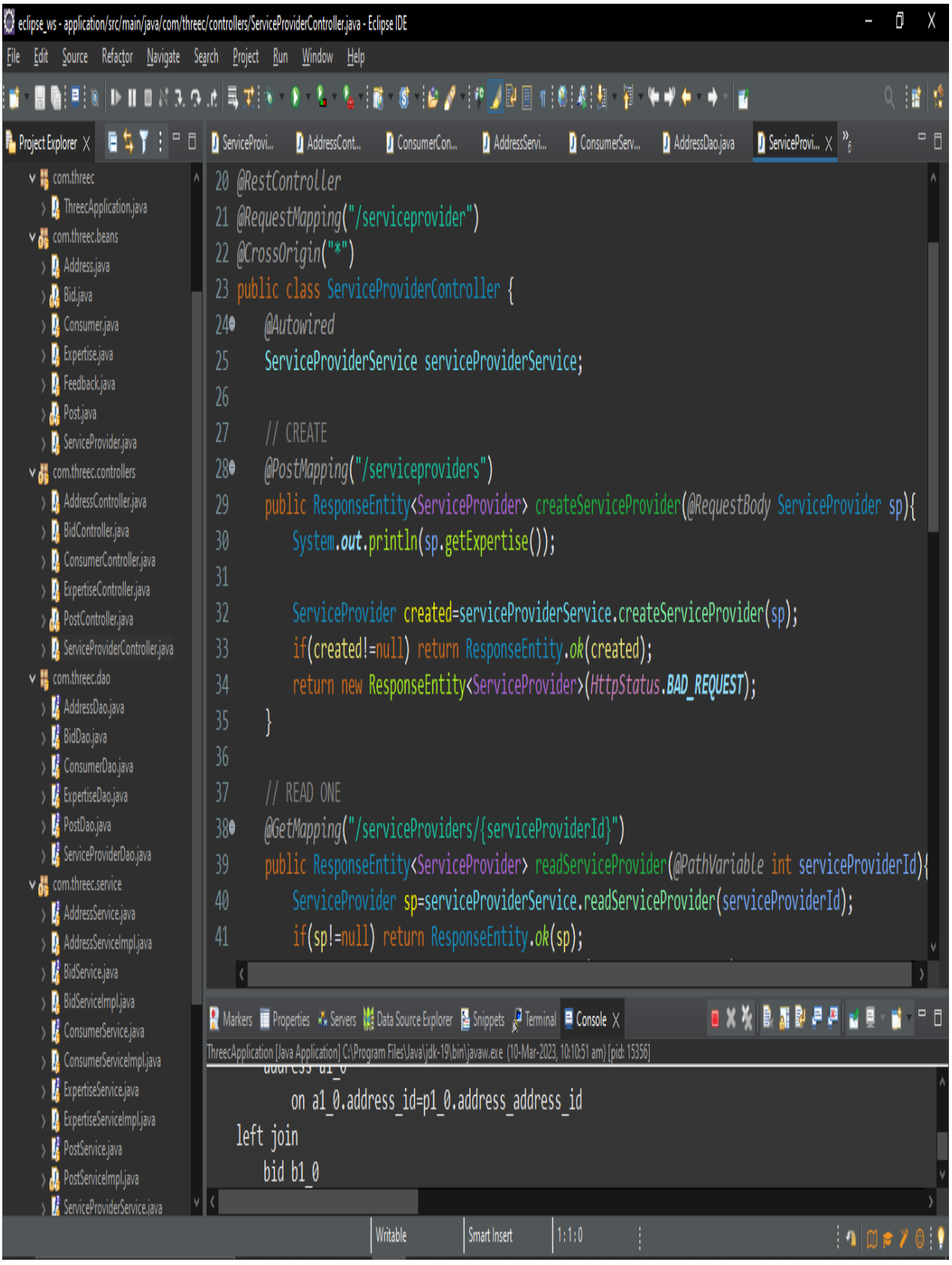

eclipse_ws - application/src/main/java/com/threec/beans/Bid.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ThreecApplication.java
Address.java
Bid.java
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
3 import com.fasterxml.jackson.annotation.JsonProperty;
16
17 @NoArgsConstructor
18 @AllArgsConstructor
19 @Getter
20 @Setter
21 @Entity
22 public class Bid {
23 @Id
24 @GeneratedValue
25 private int bidId;
26 @OneToOne(fetch = FetchType.EAGER)
27 @JsonProperty(access = Access.WRITE_ONLY)
28 private Post post;
29 @OneToOne(fetch = FetchType.LAZY)
30 @JsonProperty(access = Access.WRITE_ONLY)
31 private ServiceProvider serviceProvider;
32 private double amount;
33 private String note;
34 private boolean accepted;
35 }
36
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
21 : 8 : 561

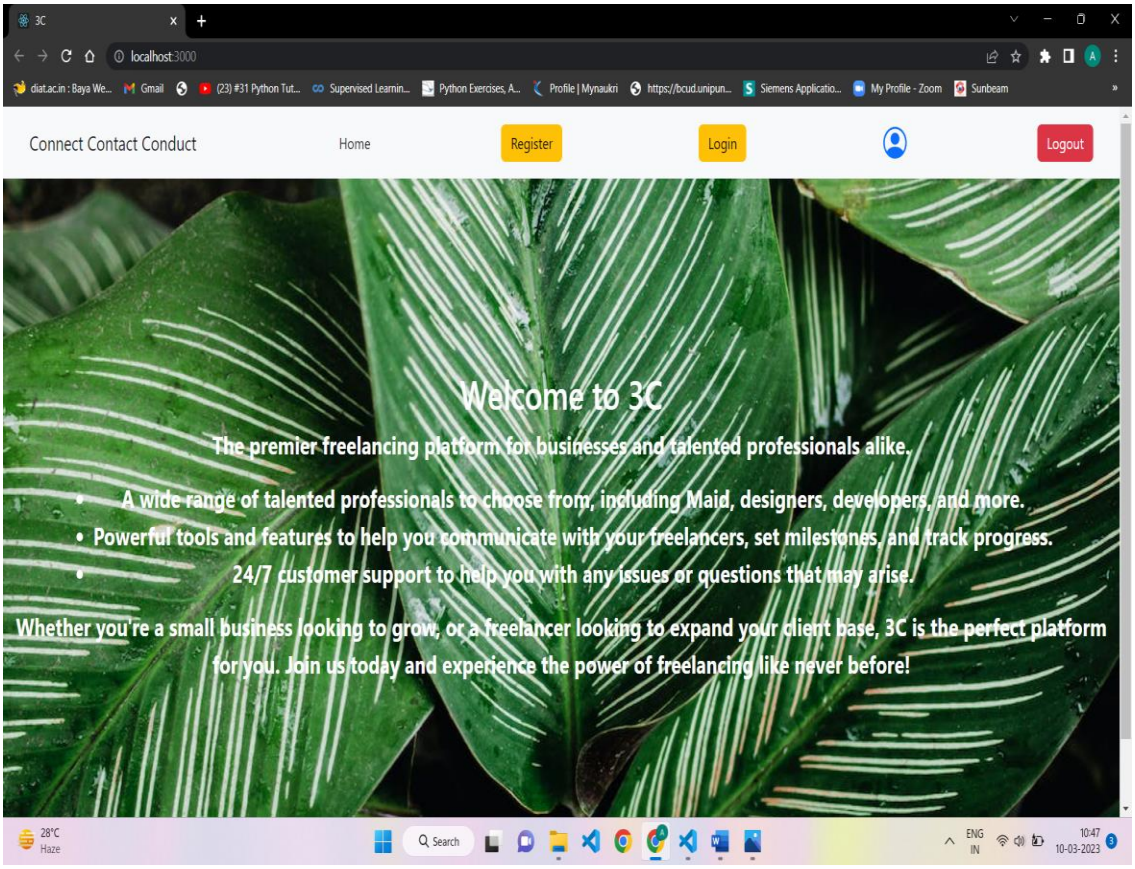

eclipse_ws - application/src/main/java/com/threec/beans/Address.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ThreecApplication.java
Address.java
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
2
3 import com.fasterxml.jackson.annotation.JsonProperty;
17
18 @NoArgsConstructor
19 @AllArgsConstructor
20 @Getter
21 @Setter
22 @Entity
23 public class Address {
24 @Id
25 @GeneratedValue
26 private int addressId;
27 @OneToOne(fetch = FetchType.LAZY, cascade = CascadeType.PERSIST)
28 @JoinColumn(name = "consumerId")
29 @JsonProperty(access = Access.WRITE_ONLY)
30 private Consumer consumer;
31 private String details;
32 private String locality;
33 private int pincode;
34 private String city;
35 }
36
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0


eclipse_ws - application/src/main/java/com/threec/ThreecApplication.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ThreecApplication.java
1 package com.threec;
2
3 import org.springframework.boot.SpringApplication;
5
6 @SpringBootApplication
7 public class ThreecApplication {
8 public static void main(String[] args) {
9 SpringApplication.run(ThreecApplication.class, args);
10 }
11 }
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
on p1_0.post_id=b1_0.post_post_id
left join
expertise c1_0
on c1_0.expertise_id=p1_0.category_expertise_id
where
Writable
Smart Insert
11 : 2 : 300

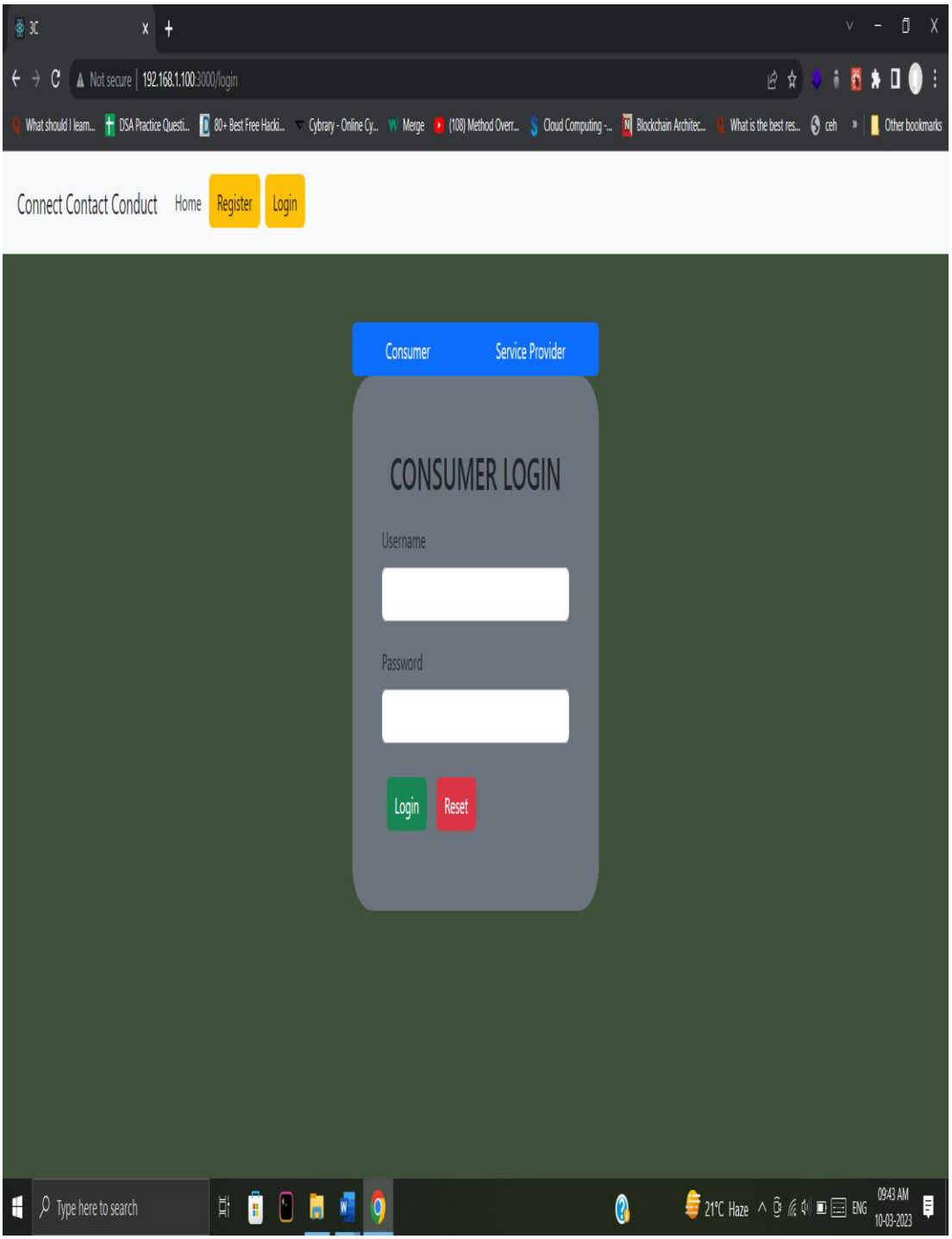

eclipse_ws - application/src/main/java/com/threec/dao/ServiceProviderDao.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ServiceProviderServiceImpl.java
src/test/java
src
ConsumerCon...
AddressServi...
ConsumerServ...
AddressDao.java
ServiceProvi...
ServiceProvi...
ServiceProvi...
1 package com.threec.dao;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
8
9 public interface ServiceProviderDao extends JpaRepository<ServiceProvider, Integer>{
10
11 @Query("select new ServiceProvider(sp.serviceProviderId, sp.username, sp.password) from ServiceF
12 ServiceProvider getLogin(@Param("n") String username);
13
14 }
15
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0


eclipse_ws - application/src/main/java/com/threec/service/ServiceProviderServiceImpl.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ServiceProviderServiceImpl.java
src/test/java
src
AddressCont...
ConsumerCon...
AddressServi...
ConsumerServ...
AddressDao.java
ServiceProvi...
ServiceProvi...
12 @Service
13 public class ServiceProviderServiceImpl implements ServiceProviderService{
14
15 @Autowired
16 ServiceProviderDao serviceProviderDao;
17
18 @Override
19 public ServiceProvider createServiceProvider(ServiceProvider sp) {
20 return serviceProviderDao.save(sp);
21 }
22
23 @Override
24 public ServiceProvider readServiceProvider(int serviceProviderId) {
25 Optional<ServiceProvider> op=serviceProviderDao.findById(serviceProviderId);
26 return op.orElse(null);
27 }
28
29 @Override
30 public List<ServiceProvider> readAll() {
31 return serviceProviderDao.findAll();
32 }
33
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0

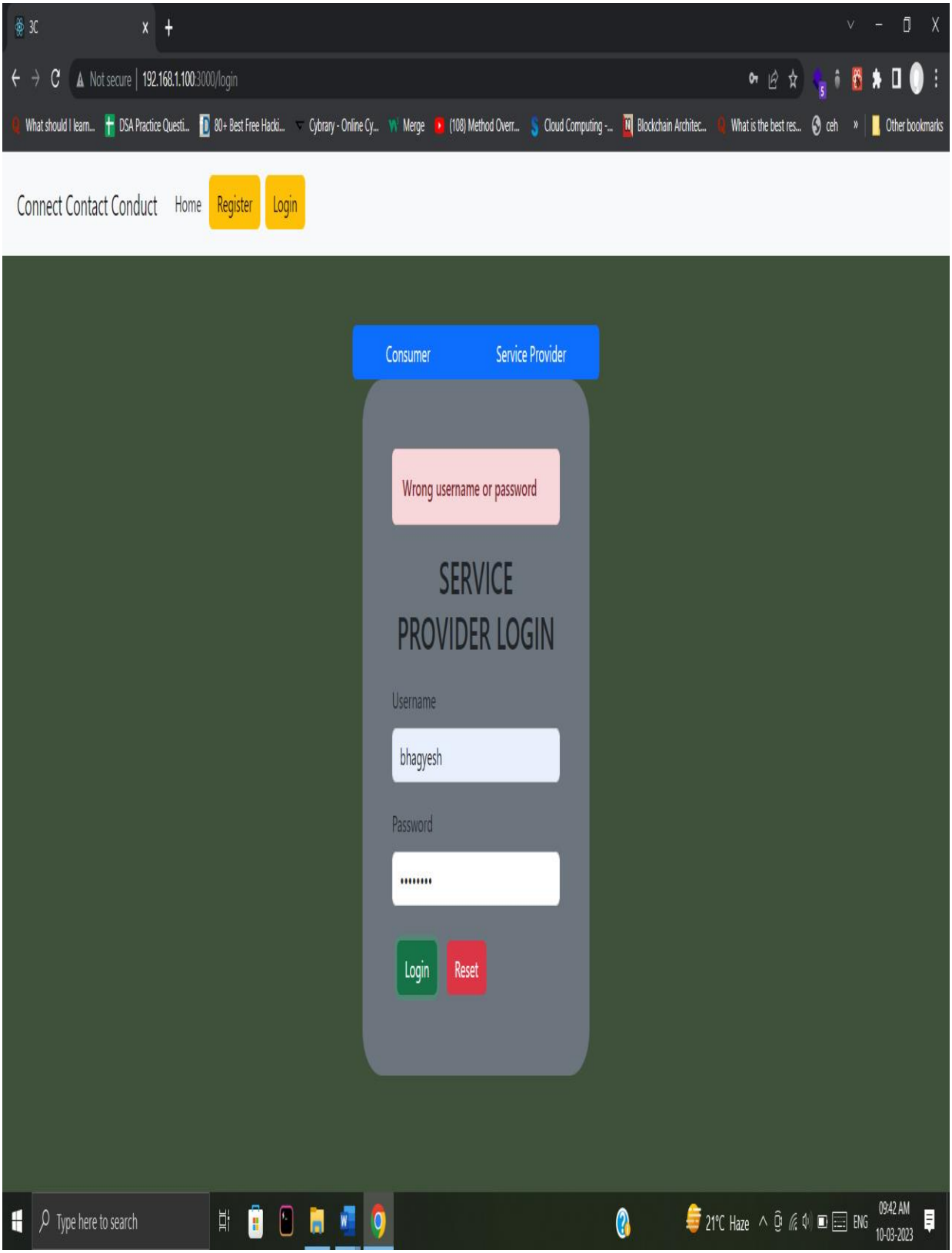

eclipse_ws - application/src/main/java/com/threec/controllers/ServiceProviderController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
com.threec
ThreecApplication.java
com.threec.beans
Address.java
Bid.java
Consumer.java
Expertise.java
Feedback.java
Post.java
ServiceProvider.java
com.threec.controllers
AddressController.java
BidController.java
ConsumerController.java
ExpertiseController.java
PostController.java
ServiceProviderController.java
com.threec.dao
AddressDao.java
BidDao.java
ConsumerDao.java
ExpertiseDao.java
PostDao.java
ServiceProviderDao.java
com.threec.service
AddressService.java
AddressServiceImpl.java
BidService.java
BidServiceImpl.java
ConsumerService.java
ConsumerServiceImpl.java
ExpertiseService.java
ExpertiseServiceImpl.java
PostService.java
PostServiceImpl.java
ServiceProviderService.java
ServiceProvi...
AddressCont...
ConsumerCon...
AddressServi...
ConsumerServ...
AddressDao.java
ServiceProvi...
20 @RestController
21 @RequestMapping("/serviceprovider")
22 @CrossOrigin("*")
23 public class ServiceProviderController {
24 @Autowired
25 ServiceProviderService serviceProviderService;
26
27 // CREATE
28 @PostMapping("/serviceproviders")
29 public ResponseEntity<ServiceProvider> createServiceProvider(@RequestBody ServiceProvider sp){
30 System.out.println(sp.getExpertise());
31
32 ServiceProvider created=serviceProviderService.createServiceProvider(sp);
33 if(created!=null) return ResponseEntity.ok(created);
34 return new ResponseEntity<ServiceProvider>(HttpStatus.BAD_REQUEST);
35 }
36
37 // READ ONE
38 @GetMapping("/serviceProviders/{serviceProviderId}")
39 public ResponseEntity<ServiceProvider> readServiceProvider(@PathVariable int serviceProviderId){
40 ServiceProvider sp=serviceProviderService.readServiceProvider(serviceProviderId);
41 if(sp!=null) return ResponseEntity.ok(sp);
Markers Properties Servers Data Source Explorer Snippets Terminal Console
ThreecApplication [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10-Mar-2023, 10:10:51 am) [pid: 15356]
on a1_0.address_id=p1_0.address_address_id
left join
bid b1_0
Writable
Smart Insert
1:1:0



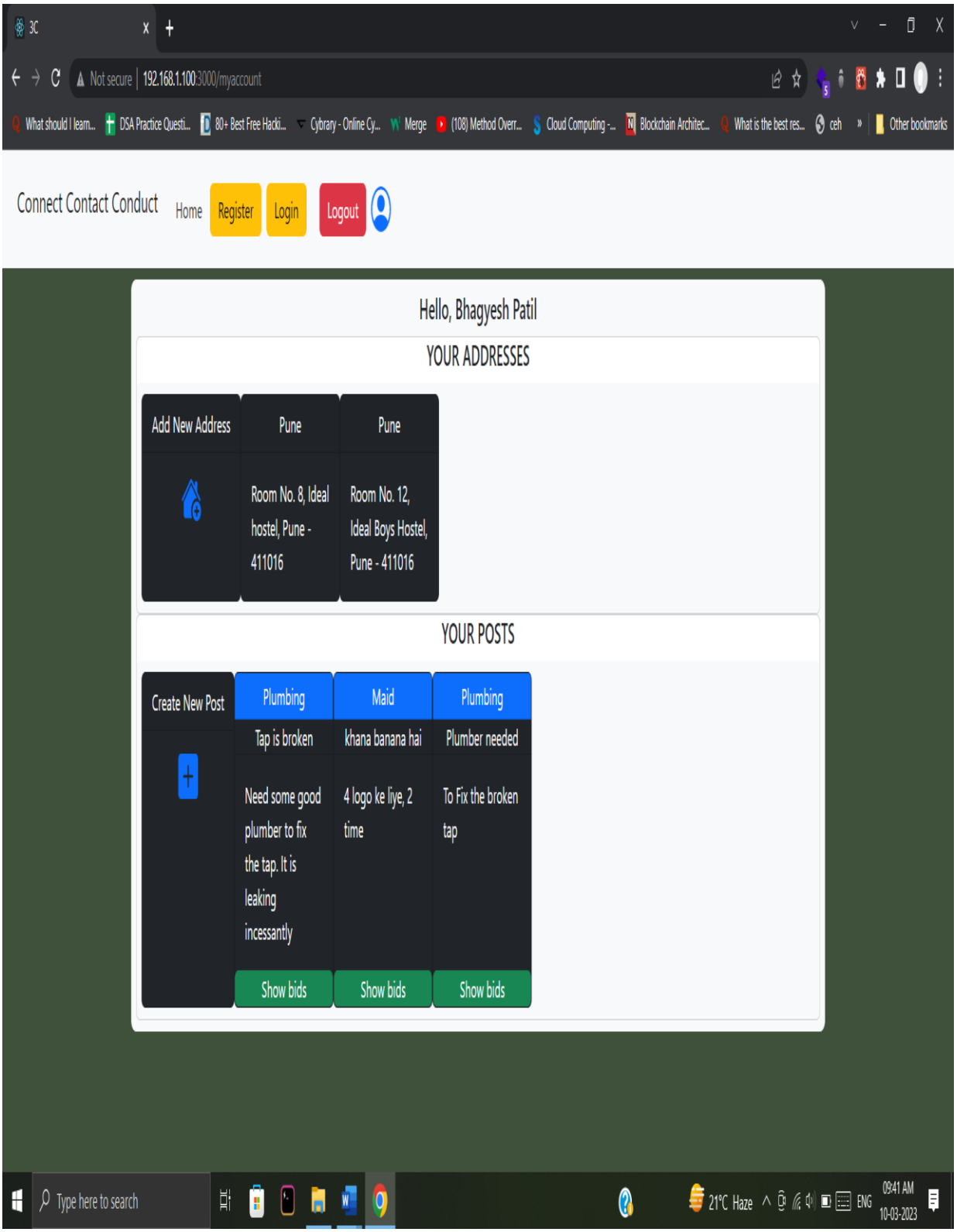
Screenshot: Home Page


Not secure | 192.168.1.100:3000/login
Connect Contact Conduct
Home
Register
Login
Consumer
Service Provider
CONSUMER LOGIN
Username
Password
Login
Reset
Type here to search
21°C Haze
ENG
09:43 AM
10-03-2023

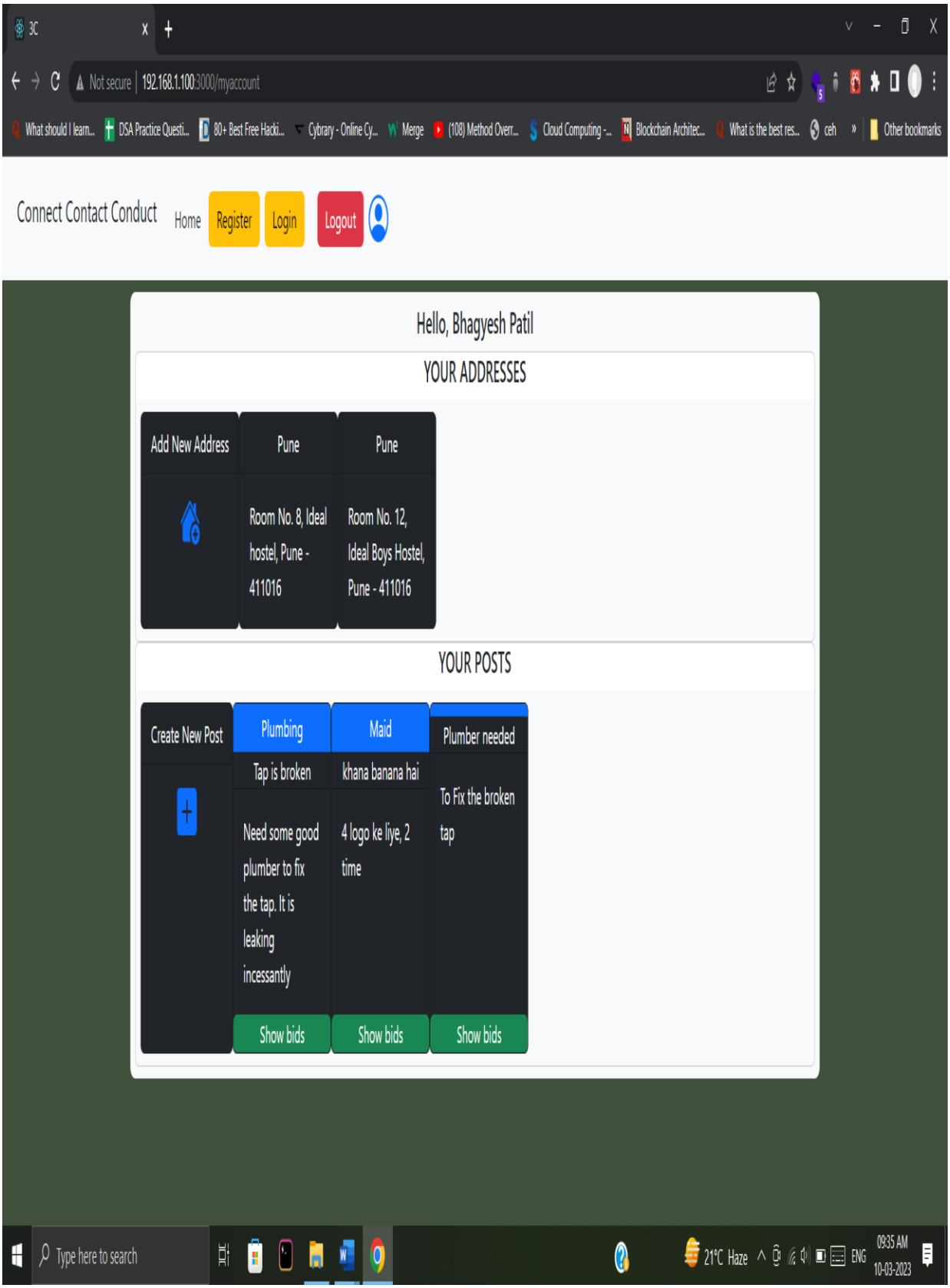
Screenshot:


Not secure | 192.168.1.100:3000/login
Connect Contact Conduct
Home
Register
Login
Consumer
Service Provider
Wrong username or password
SERVICE PROVIDER LOGIN
Username
bhagyesh
Password
Login
Reset
Type here to search
21°C Haze
ENG
09:42 AM
10-03-2023

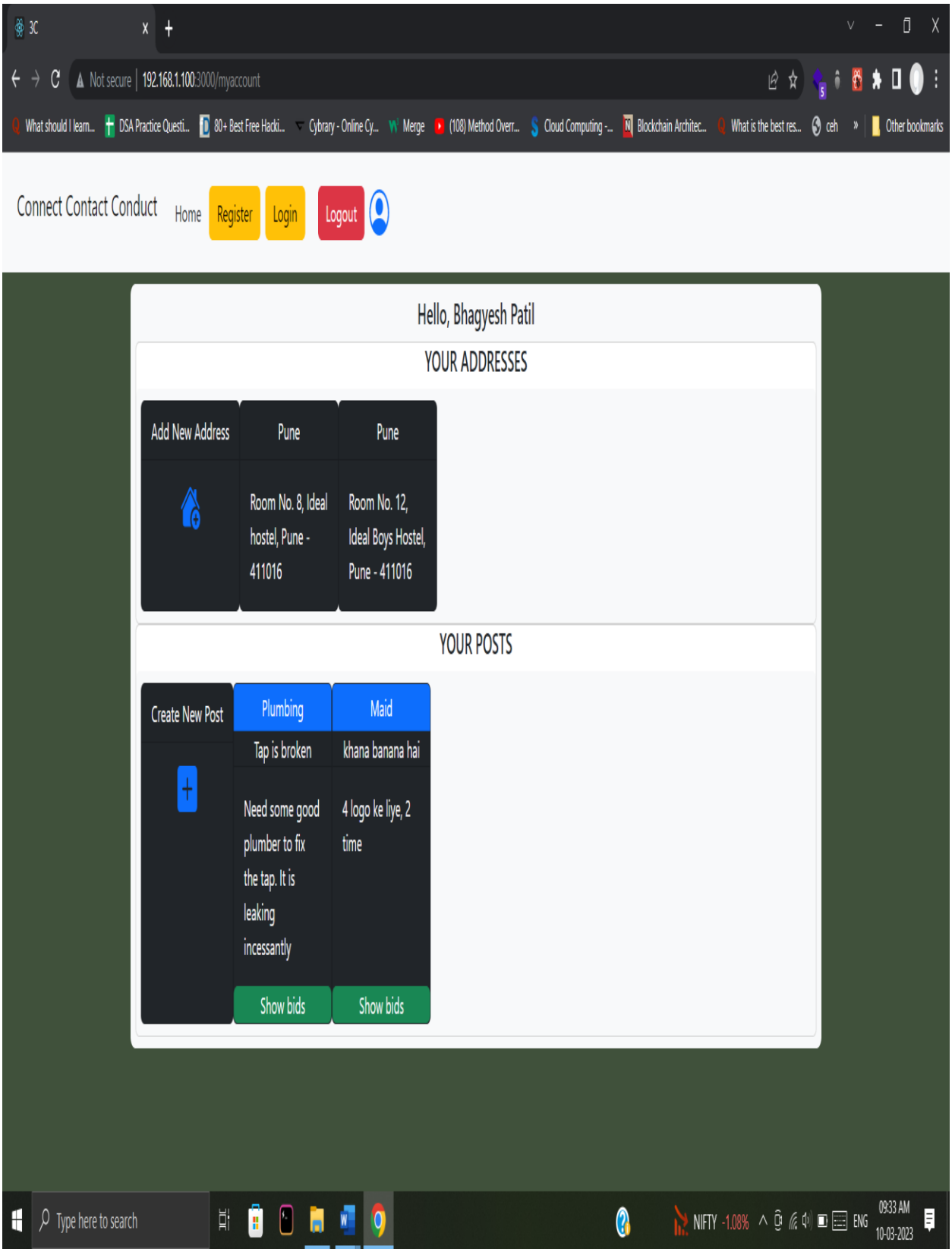
Screenshot:


Not secure | 192.168.1.100:3000/myaccount
Connect Contact Conduct
Home
Register
Login
Logout
Hello, Bhagyesh Patil
YOUR ADDRESSES
Add New Address
Pune
Room No. 8, Ideal hostel, Pune - 411016
Pune
Room No. 12, Ideal Boys Hostel, Pune - 411016
YOUR POSTS
Create New Post
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Show bids
Maid
khana banana hai
4 logo ke liye, 2 time
Show bids
Plumbing
Plumber needed
To Fix the broken tap
Show bids
Type here to search
21°C Haze
ENG
09:41 AM
10-03-2023

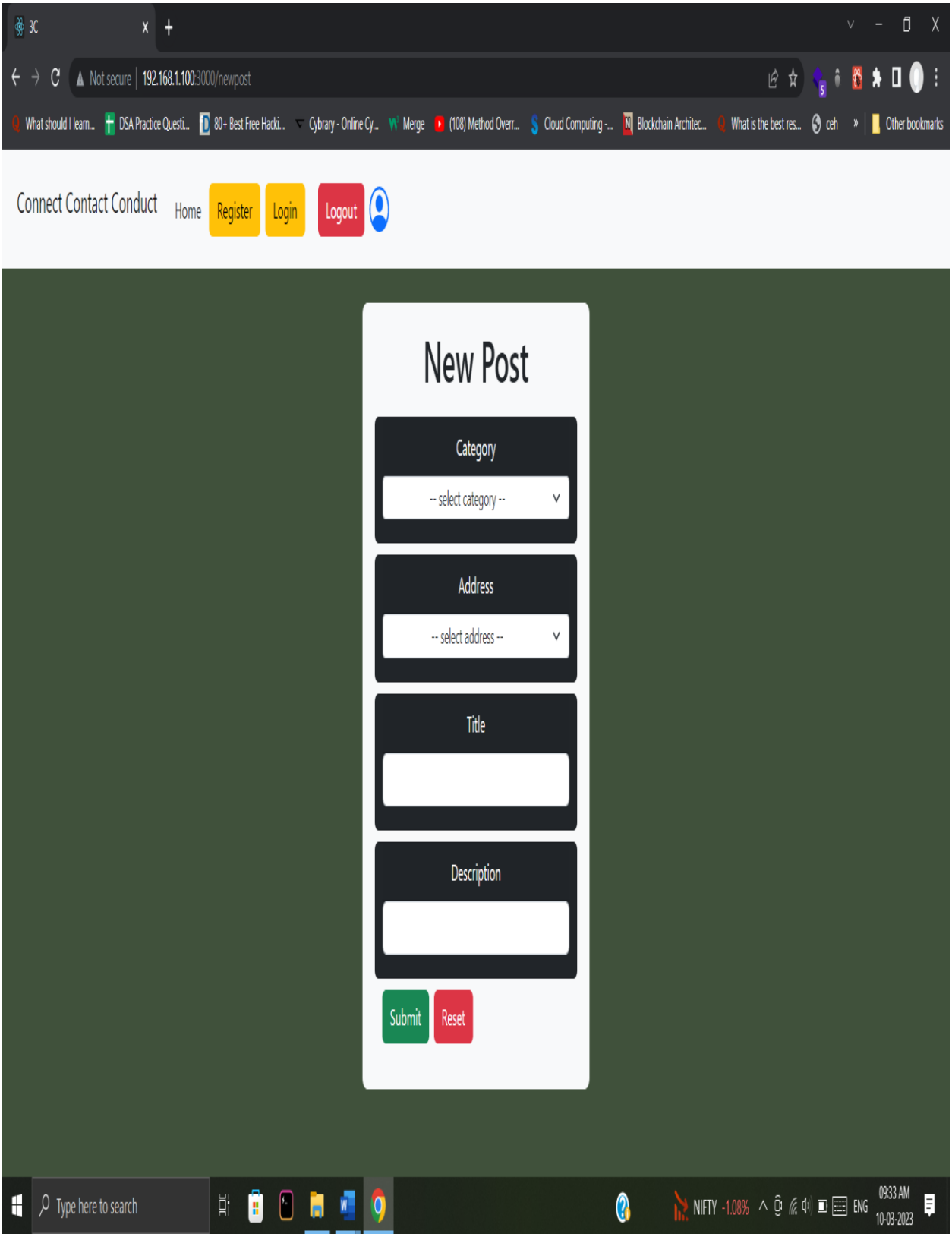
Screenshot:


Not secure | 192.168.1.100:3000/myaccount
Connect Contact Conduct
Home
Register
Login
Logout
Hello, Bhagyesh Patil
YOUR ADDRESSES
Add New Address
Pune
Room No. 8, Ideal hostel, Pune - 411016
Pune
Room No. 12, Ideal Boys Hostel, Pune - 411016
YOUR POSTS
Create New Post
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Show bids
Maid
khana banana hai
4 logo ke liye, 2 time
Show bids
Plumber needed
To Fix the broken tap
Show bids
Type here to search
21°C Haze
ENG
09:35 AM
10-03-2023

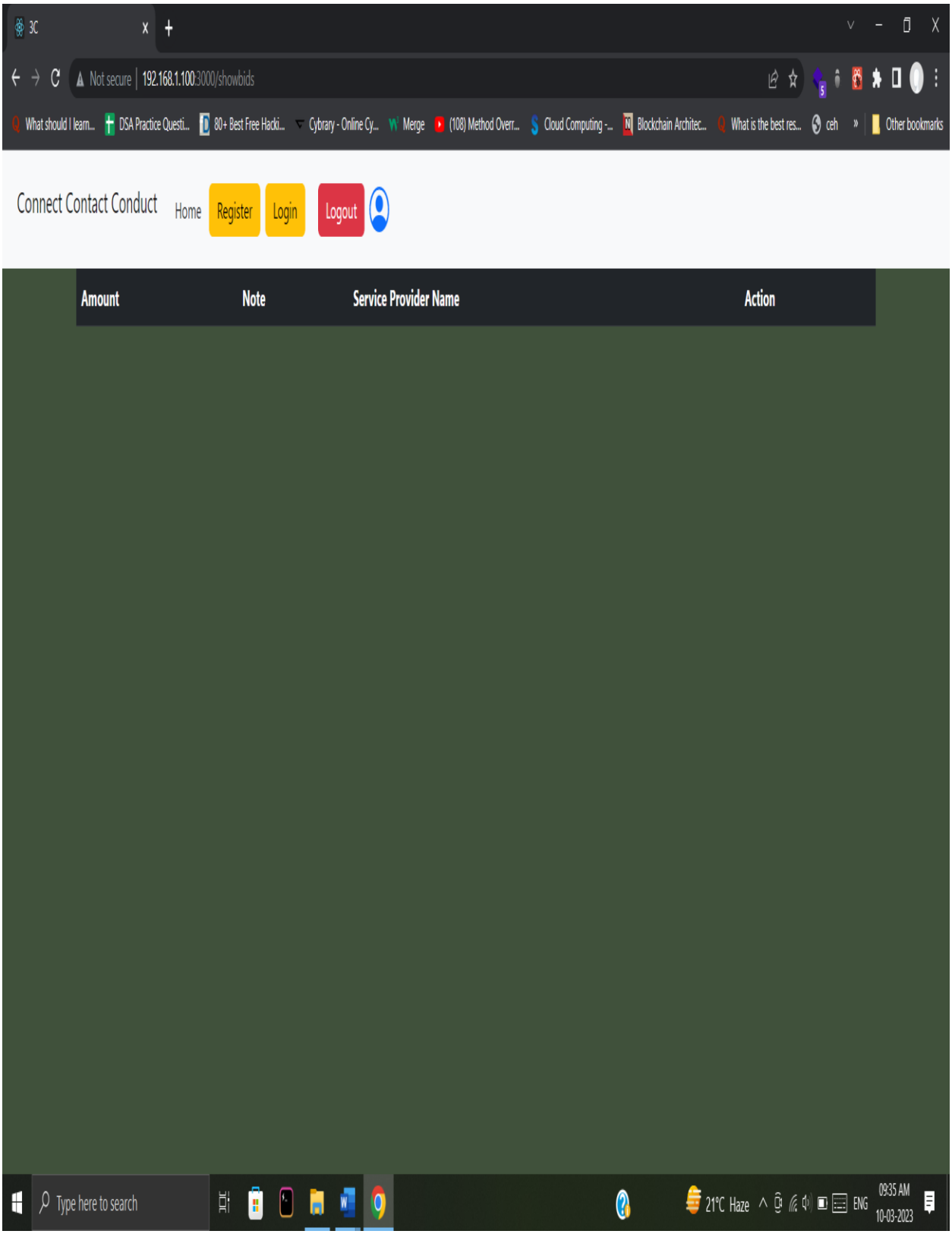
Screenshot:


Not secure | 192.168.1.100:3000/myaccount
Connect Contact Conduct
Home
Register
Login
Logout
Hello, Bhagyesh Patil
YOUR ADDRESSES
Add New Address
Pune
Room No. 8, Ideal hostel, Pune - 411016
Pune
Room No. 12, Ideal Boys Hostel, Pune - 411016
YOUR POSTS
Create New Post
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Show bids
Maid
khana banana hai
4 logo ke liye, 2 time
Show bids
Type here to search
NIFTY -1.08%
ENG
09:33 AM
10-03-2023

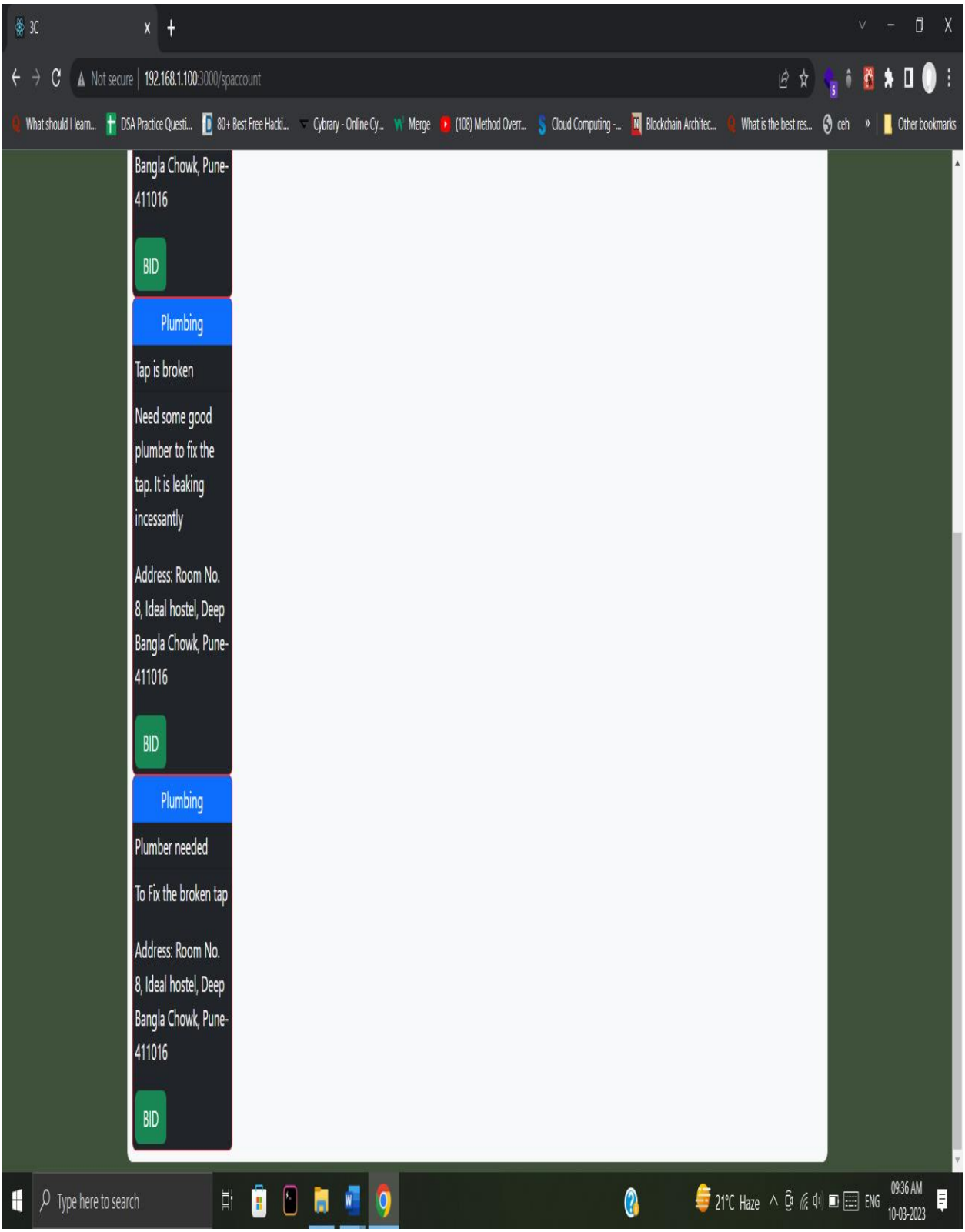
Screenshot:


Not secure | 192.168.1.100:3000/newpost
Connect Contact Conduct
Home
Register
Login
Logout
New Post
Category
-- select category --
Address
-- select address --
Title
Description
Submit
Reset
Type here to search
NIFTY -1.08%
ENG
09:33 AM
10-03-2023

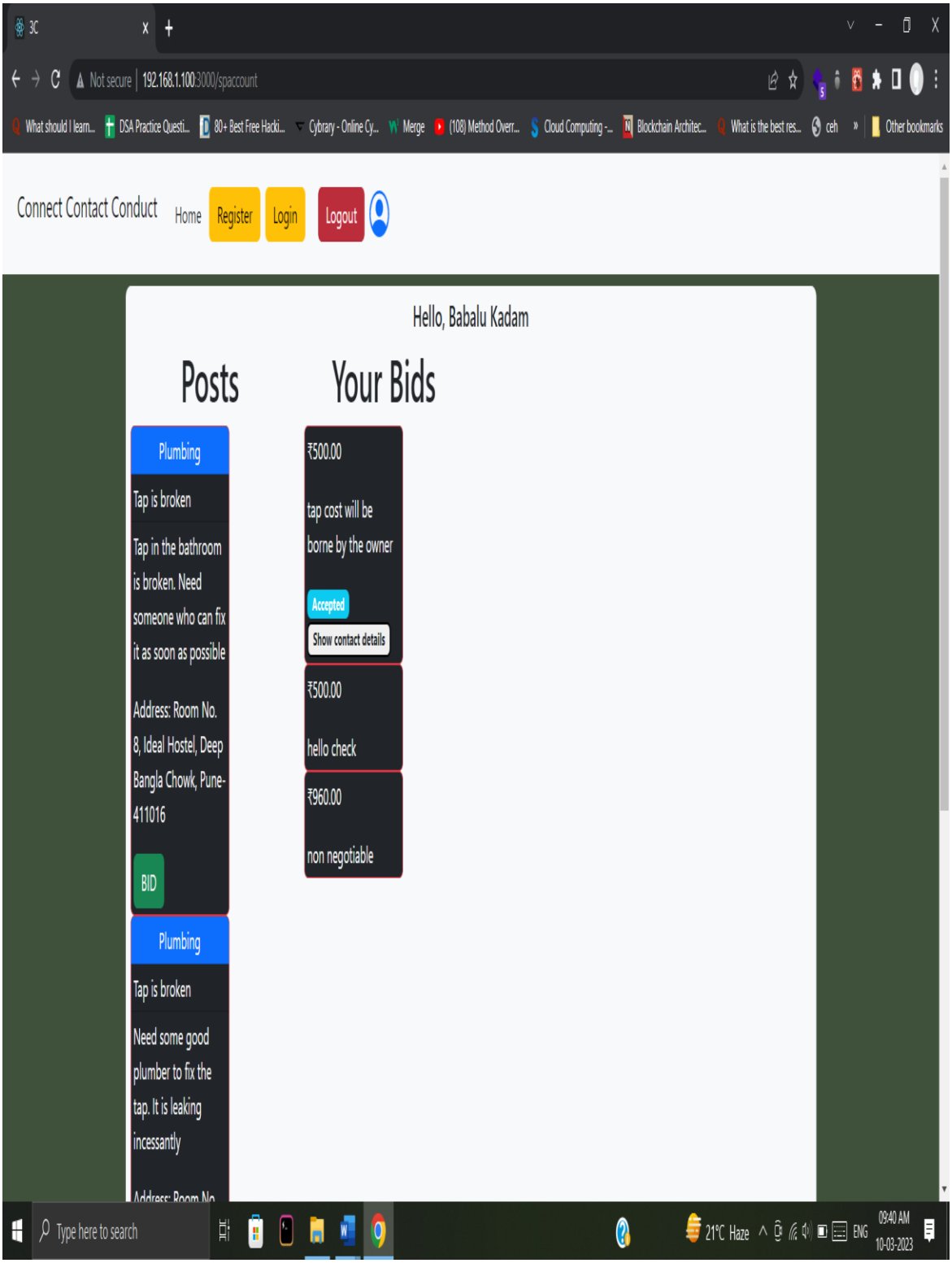
Screenshot:


3C
Not secure | 192.168.1.100:3000/showbids
What should I learn...
DSA Practice Questi...
80+ Best Free Hacki...
Cybrary - Online Cy...
Merge
(108) Method Overr...
Cloud Computing -...
Blockchain Architec...
What is the best res...
ceh
Other bookmarks
Connect Contact Conduct
Home
Register
Login
Logout
Amount
Note
Service Provider Name
Action
Type here to search
21°C Haze
ENG
09:35 AM
10-03-2023

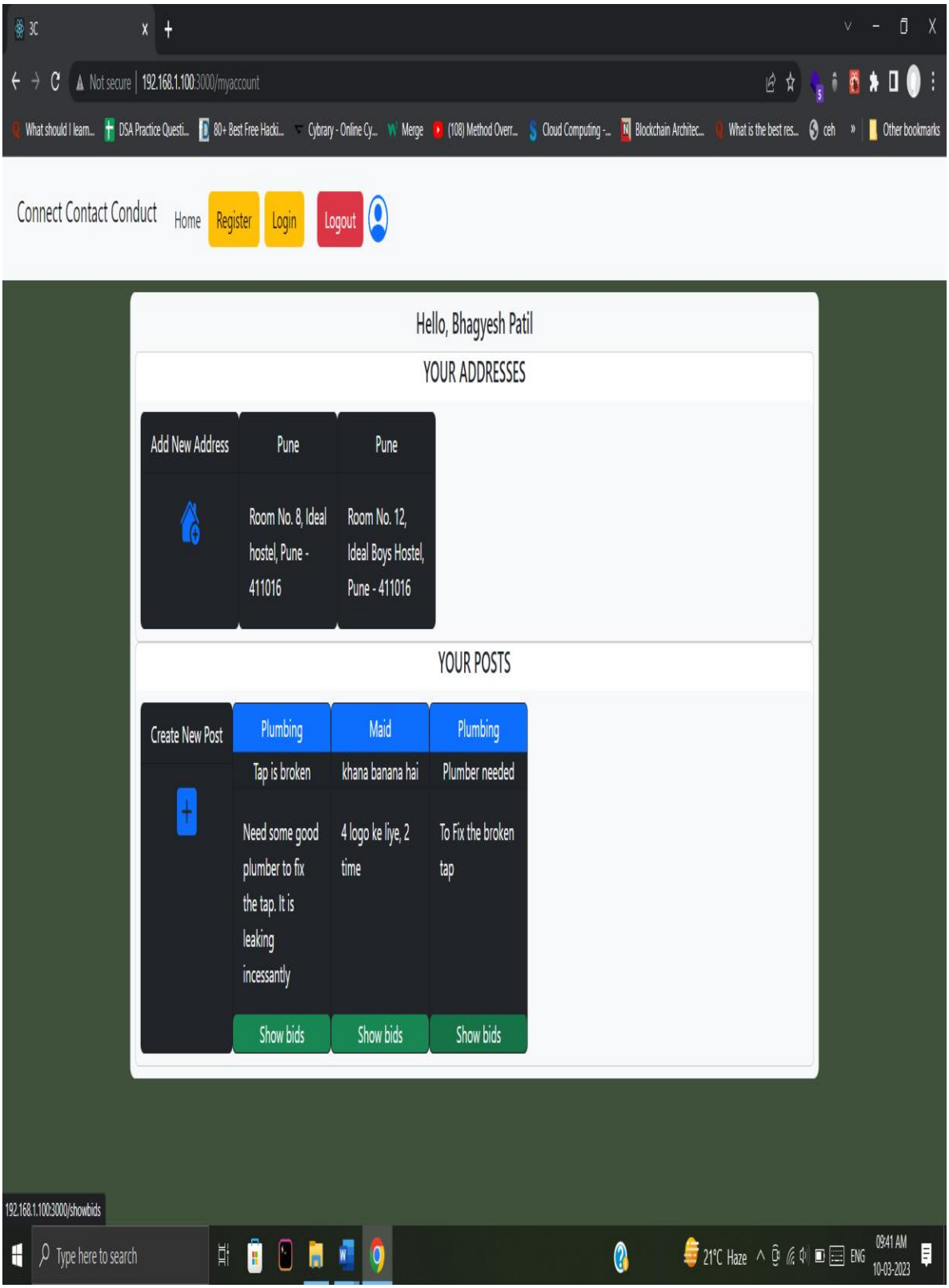
Screenshot:


3C
Not secure | 192.168.1.100:3000/spaccount
What should I learn...
DSA Practice Questi...
80+ Best Free Hacki...
Cybrary - Online Cy...
Merge
(108) Method Overr...
Cloud Computing -...
Blockchain Architec...
What is the best res...
ceh
Other bookmarks
Bangla Chowk, Pune-411016
BID
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Address: Room No. 8, Ideal hostel, Deep Bangla Chowk, Pune-411016
BID
Plumbing
Plumber needed
To Fix the broken tap
Address: Room No. 8, Ideal hostel, Deep Bangla Chowk, Pune-411016
BID
Type here to search
21°C Haze
ENG
09:36 AM
10-03-2023

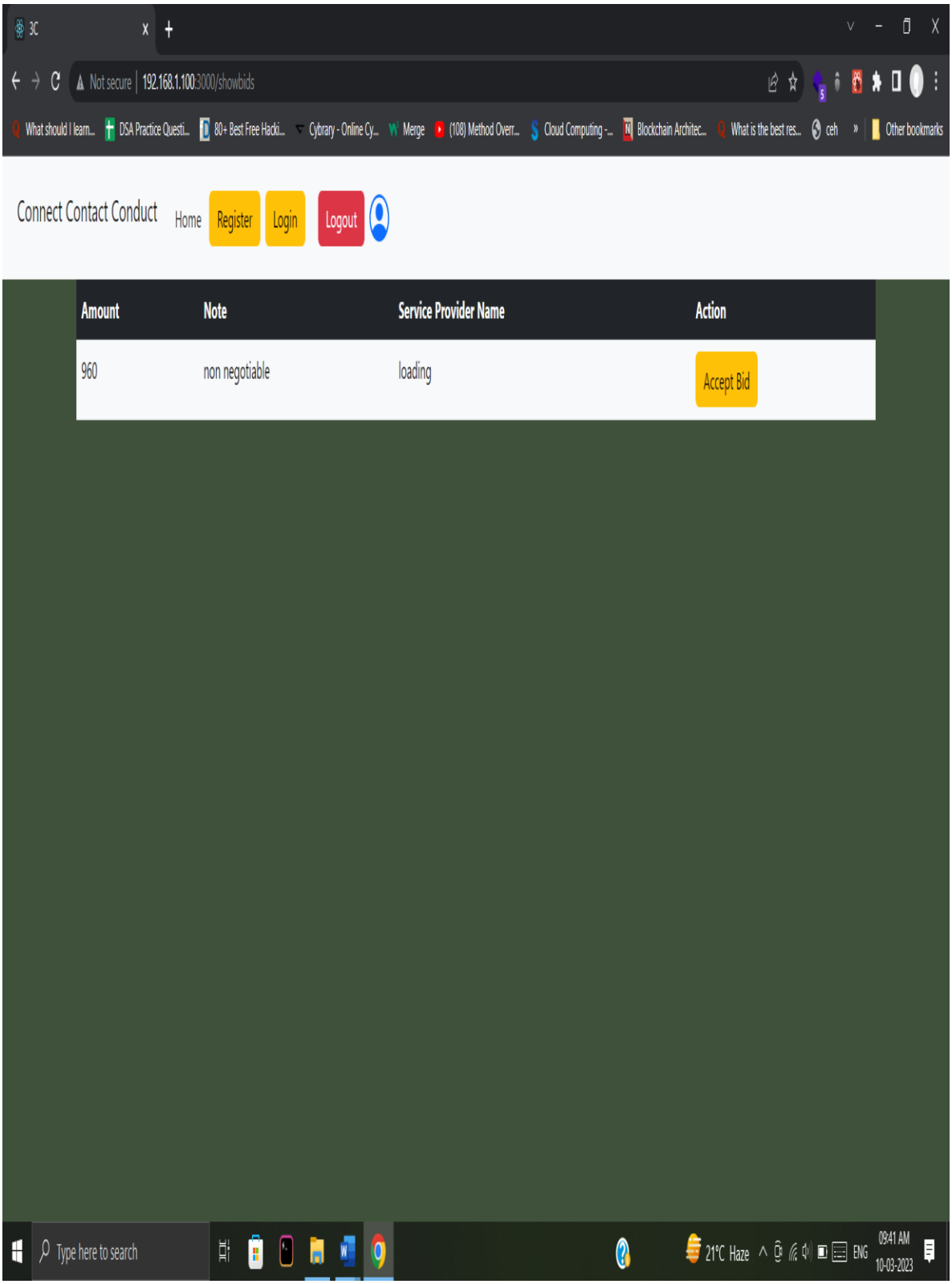
Screenshot:


Not secure | 192.168.1.100:3000/spaccount
Connect Contact Conduct
Home
Register
Login
Logout
Hello, Babalu Kadam
Posts
Your Bids
Plumbing
Tap is broken
Tap in the bathroom is broken. Need someone who can fix it as soon as possible
Address: Room No. 8, Ideal Hostel, Deep Bangla Chowk, Pune-411016
BID
₹500.00
tap cost will be borne by the owner
Accepted
Show contact details
₹500.00
hello check
₹960.00
non negotiable
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Type here to search
21°C Haze
ENG
09:40 AM
10-03-2023

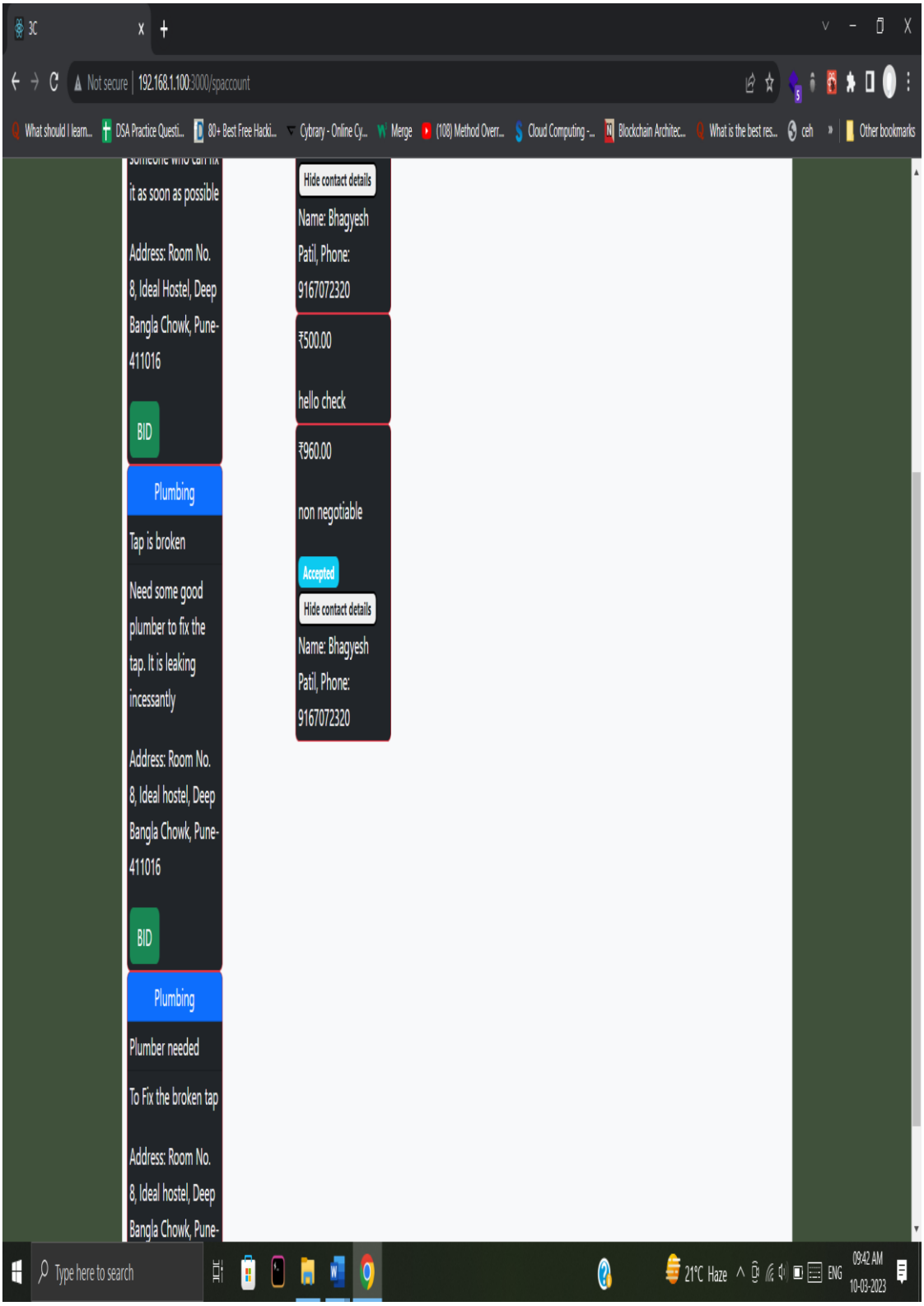
Screenshot:


Not secure | 192.168.1.100:3000/myaccount
Connect Contact Conduct
Home
Register
Login
Logout
Hello, Bhagyesh Patil
YOUR ADDRESSES
Add New Address
Pune
Room No. 8, Ideal hostel, Pune - 411016
Pune
Room No. 12, Ideal Boys Hostel, Pune - 411016
YOUR POSTS
Create New Post
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Show bids
Maid
khana banana hai
4 logo ke liye, 2 time
Show bids
Plumbing
Plumber needed
To Fix the broken tap
Show bids
192.168.1.100:3000/showbids
Type here to search
21°C Haze
ENG
09:41 AM
10-03-2023

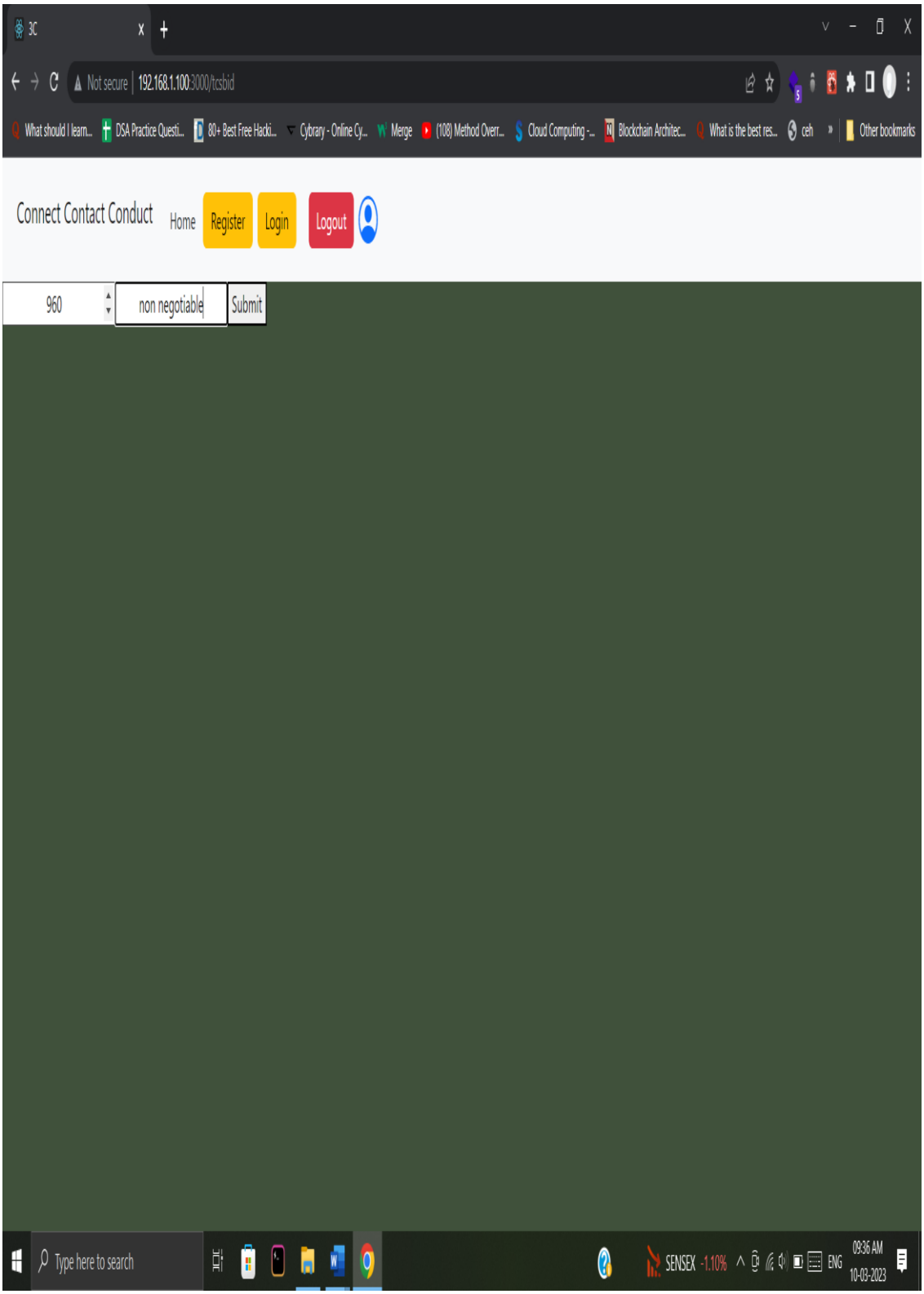
Screenshot:


3C
Not secure | 192.168.1.100:3000/showbids
What should I learn...
DSA Practice Questi...
80+ Best Free Hacki...
Cybrary - Online Cy...
Merge
(108) Method Overr...
Cloud Computing -...
Blockchain Architec...
What is the best res...
ceh
Other bookmarks
Connect Contact Conduct
Home
Register
Login
Logout
Amount
Note
Service Provider Name
Action
960
non negotiable
loading
Accept Bid
Type here to search
21°C Haze
ENG
09:41 AM
10-03-2023

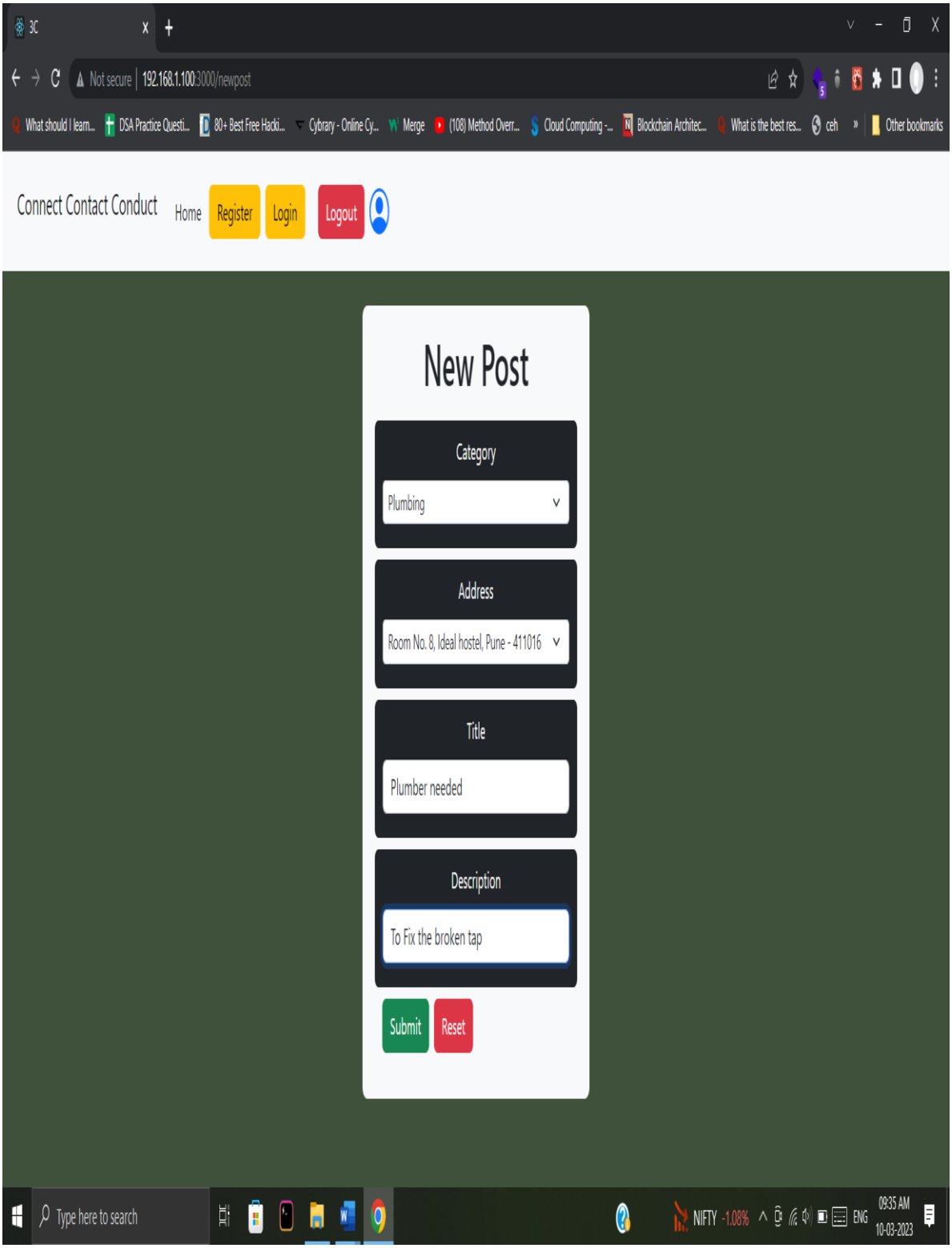
Screenshot:


Not secure | 192.168.1.100:3000/spaccount
it as soon as possible
Address: Room No. 8, Ideal Hostel, Deep Bangla Chowk, Pune-411016
BID
Plumbing
Tap is broken
Need some good plumber to fix the tap. It is leaking incessantly
Address: Room No. 8, Ideal hostel, Deep Bangla Chowk, Pune-411016
BID
Plumbing
Plumber needed
To Fix the broken tap
Address: Room No. 8, Ideal hostel, Deep Bangla Chowk, Pune-
Hide contact details
Name: Bhagyesh Patil, Phone: 9167072320
₹500.00
hello check
₹960.00
non negotiable
Accepted
Hide contact details
Name: Bhagyesh Patil, Phone: 9167072320
Type here to search
21°C Haze
09:42 AM
10-03-2023

Screenshot:


Not secure 192.168.1.100:3000/tcsbid
What should I learn...
DSA Practice Questi...
80+ Best Free Hacki...
Cybrary - Online Cy...
Merge
(108) Method Overr...
Cloud Computing -...
Blockchain Architec...
What is the best res...
ceh
Other bookmarks
Connect Contact Conduct
Home
Register
Login
Logout
960
non negotiable
Submit
Type here to search
SENSEX -1.10%
ENG
09:36 AM
10-03-2023

Screenshot:


Not secure | 192.168.1.100:3000/newpost
Connect Contact Conduct
Home
Register
Login
Logout
New Post
Category
Plumbing
Address
Room No. 8, Ideal hostel, Pune - 411016
Title
Plumber needed
Description
To Fix the broken tap
Submit
Reset
Type here to search
NIFTY -1.08%
ENG
09:35 AM
10-03-2023

Screenshot: