**INSTANT HIRE**

**PG Diploma in Advanced Computing**

By

**Harshada Kerkar (PL), 200240320039**

**Shankar Lad, 200240320104**

**Prathamesh Patil, 200240320075**

**Meghana Mahajan, 200240320056**

**Shubham Naik, 200240320117**

**Name of the Project Guide**

Poorti Sharma

Aishwarya Kale

**Name of the Faculty Project Guide**

Sohan More

****

**Centre for Development of Advanced Computing**

**Kharghar**

**February 2021**

**CERTIFICATE**

This is to certify that the project entitled **“Instant Hire”** is a bonafied work of Harshada Kerkar (PL) (200240320039)Shankar Lad (200240320104) Prathamesh Patil (200240320075), Meghana Mahajan (200240320056), and Shubham Naik (200240320117). Submitted to C-DAC Mumbai in partial fulfilment of the requirement for the award of the Post Graduate Diploma in Advanced Computing.

Poorti Sharma

Aishwarya Kale Sohan More

**Supervisor/Guide Faculty Supervisor/Guide**

**Declaration**

I declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----------------------------------------

Harshada Kerkar (PL) (200240320039)

-----------------------------------------

Shankar Lad (200240320104)

-----------------------------------------

Prathamesh Patil (200240320075)

-----------------------------------------

Meghana Mahajan (200240320056)

-----------------------------------------

Shubham Naik (200240320117)

Date:

**Abstract**

The on demand home service system is incredibly useful for everybody who wants to urge home services like plumbing, electronic repair, chef, car driver ,painter, event manager and electrical maintenance. When an individual relocating from one area to a different because now a day’s everyone wants to save lots of time and shot out their problems within time with none problem. Therefore, online home services are very beneficial for people. There are only two users in our system, first is Home Service providers and therefore the other may be a user. Home service providers have a crucial role within the project he/she can register with this website by mentioning their role and adds.

The small about them by providing their contact number while the user can see an inventory of home services and get in touch with them as per their requirements. The web home service project consists of the many categories and services as mentioned before. Users who are in need of services can register with this website and look for service providers by mentioning the situation. The service provider’s there in particular locations are listed to user with contact number and therefore the user can contact them. By this users can easily avail the needed home services with none difficulty and delay.

The main objective of the on demand home services is providing the house services by one click. This paper discusses about the web home services, several services provided and therefore the method of ordering and delivery of services. On demand home services system are often accessed by registered users to seem for household services through an insightful web application. The event of web based online system helps in determining household services and collaborating interface to look the services. The system also acknowledges the confirmation of services chosen by the users

**Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Contents** | **Page No.** |
| **1** | **INTRODUCTION** | **11** |
|  | **1.1 Description** | **11** |
|  | **1.2 Problem Formulation** | **12** |
|  | **1.3 Motivation** | **12** |
|  | **1.3 Proposed Solution** | **12** |
|  | **1.4 Scope of the project** | **13** |
| **2** | **REVIEW OF LITERATURE** | **14** |
| **3** | **SYSTEM ANALYSIS** | **15** |
|  | **3.1 Functional Requirements** | **15** |
|  | **3.2 Non Functional Requirements** | **16** |
|  | **3.3 Specific Requirements** | **16** |
| **4** | **ANALYSIS MODELING** | **17** |
|  | **4.1 Use-Case Diagrams and description**  **4.2 Activity Diagrams**  **4.3 Class Diagram**  **4.4 Sequence Diagram** | **17** |
|  | **4.5 Timeline Chart** | **25** |
| **5** | **DESIGN** | **26** |
|  | **5.1 Data Modeling *(E-R Model)*** | **26** |
|  | **5.2 User Interface Design** | **27** |
| **6** | **IMPLEMENTATION** | **33** |
|  | **6.1 Algorithms / Methods Used** | **34** |
|  | **6.2 Working of the project** | **35** |
| **7** | **TESTING** | **36** |
|  | **7.1 Type of Testing used** | **36** |
|  | **7.2 Test cases** | **37** |
| **8** | **RESULTS AND DISCUSSIONS** | **38** |
| **9** | **CONCLUSIONS & FUTURE SCOPE** | **41** |

Literature Cited

Acknowledgments

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Figure Caption** | **Page No.** |
| 4.1 | User Case Diagram:a.1 | 17 |
| 4.1 | User Case Diagram:a.2 | 18 |
| 4.2 | Class Diagram | 19 |
| 4.3.1 | Activity Diagram | 20 |
| 4.3.2 | Activity Diagram | 21 |
| 4.3.3 | Activity Diagram | 22 |
| 4.4.1 | Sequence Diagram | 23 |
| 4.4.2 | Sequence Diagram | 24 |
| 4.4.3 | Sequence Diagram | 25 |
| 5.0 | Gantt Chart | 25 |
| 5.1 | Data Modeling | 26 |
| 5.2.1 | Home Page | 27 |
| 5.2.2 | Login Page | 28 |
| 5.2.3 | Register Page | 29 |
| 5.2.4 | Customer Details Page | 29 |
| 5.2.5 | Order List Page | 30 |
| 5.2.6 | Notification List Page | 30 |
| 5.2.7 | Employee List Page | 31 |
| 5.2.8 | Customer Details Page | 31 |
| 5.2.9 | Order Details Page | 32 |
| 8.1 | Reject Page | 31 |
| 8.2 | Hire Page | 32 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Title** | **Page No.** |
| 7.2 | Test Cases | 37 |

**Chapter 1**

**INTRODUCTION**

Instant-Hire is the platform that connects consumers and professional startup, small business service providers. In this, Customers can search the services based on their own criteria and get their job done by the professionals. It is the best way that a small business can reach with the seekers through our application and connect with the local audience by showing their services online.

It is all-in-one platform for various at-home services. The application can be used to hire professionals for many at-home services like Plumber, Carpenter, Chefs, Drivers, Event-planners, Painters etc. simply it allows the users to connect with skilled and experienced professionals for at-home services.

## 1.1 Description

* The main objective of our project is to provide job to unemployed, talented, trustworthy people and also to provide instant services to our customers.
* In this project customer can hire employee according to their requirements and according to their time to get their job done.
* The application has two main fold functioning:

**User End:**

* The user can use the app if they are looking for a professional to perform a specific in-house service.
* They can choose any service from the list of services mentioned.
* Once they choose a particular service, a pop up will come which contains various professionals list and their rate of working per hour. User can select the employee according to their budget.
* After this, they are asked to choose suitable date and time.
* The app allows the user to check the profile of the professional to see their qualities.
* User can also rate and post a review of the service.

**Service Provider End:**

* The service provider can be listed only after several background checking such as NOC certificate, Aadhar Number etc.
* Once the user selects a service, the service provider can accept or reject the offer according to his willing.
* The service provider can only charge the customer according to his rate of working per hour including all the taxes.
* He can reply to customer reviews/feedback.

**Admin Panel Features:**

* Full Admin Control Panel.
* Can see list of all Professional from various categories.
* Can see list of all customers’ orders in the orders list.
* Can Add or Delete the Employees and Customers.
* Can set the prices of professionals according to feedback/reviews of the customers.

**1.2 Problem Formulation:**

For the majority of the people, it is hard to find the trustworthy and experienced professional for their in-house work. Professionals are also struggling to find work. Therefore, to solve these two problems, our application platform known as Instant-Hire can be found as a medium to resolve these problem. Through our platform user can find professionals for their in-house work through various categories of our application and also the professionals can also find some work.

* 1. **Need of Project:**
* Now a day it is very difficult to find the trustworthy and experienced professionals of various categories at one single place so our platform can provide the services to the users as well as professionals.
* Instant-Hire provides a platform that allows skilled and experienced professionals to connect with the users looking for specific service.
* It enables users to find any service professionals like Plumber, Painter, Event-Planner, and Carpenters etc. which are listed on our platform.
* Our platform is the best way that a small business can reach with the seekers through our application and connect with the local audience by showing their services online.
* Instant-Hire provides a platform that allows skilled and experienced professionals to connect with the users looking for specific service.
  1. **Project future scope:**
* Now a day it’s very hard to search trustworthy employee and searching job our project can easily provide services to users and job to jobless people.
* Our project consists of lots of category like divers, plumber, painter, even-planner, tutor, designer, web designer and many more.
* We can empower millions of service professionals across the India to deliver services at home.
* Dual way verification for working hours.

**Chapter 2**

**Review of Literature**

There are many online home service systems in existence which are discussed briefly in this section.

Urban Pro is the framework which initially began their online help for connecting the scholars with the mentors, trainers and institutes [8]. This was one among the explanations for the emerging of providing the web domestic services.

Time saverz is one among the web home service system where the customer has given rewards for the services offered and a refund if the customer isn't satisfied with the services. This service is provided in Delhi, Noida, Gurgaon, Hyderabad, Bangalore, Pune, Mumbai and Chennai [9]

Zimmber has provided the house services but they need enlisted the providers in order that the purchasers can rest their worries. This application provides the services only within the urban cities like Pune and Bangalore. This system acts as a platform not just for offering services but also for the hiring of professionals [10][12].

**Reference:**

<https://stackoverflow.com/><https://expressjs.com/>

<https://angular.io/>

<https://material.angularjs.org/latest/>

<https://nodejs.org/en/>

<https://www.urbancompany.com/delhi-ncr>

<https://www.taskrabbit.com/>

<https://mrusta.com/>

<https://bro4u.com/list-as-partner>

**Chapter 3**

**System Analysis**

## **3.1 Functional Requirements**

## **3.1.1 Login of Admin**

* The system will allow the admin to view Employees and customers.
* The system will allow the admin to search employee from Employee List.
* The system will allow the admin to give permission.
* The system will allow the admin to delete employee.
* The system will allow the admin to give rating to employee.

## **3.1.2 Login and Register of Customer**

* The system will allow Customer to Register.
* The system will allow Customer to select Employee from Employee List according to requirements.
* The system will allow Customer to hire appropriate employee.
* The system will allow the Customer to see Charges of Employee with respect to rating.
* The system will allow Customer to update own details

**3.1.3 Login and Register of Employee**

* The system will allow Employee to Register
* The system will allow Employee to The system will allow Employee to update own details.
* The system will Employee to accept or reject Customer request.
* The system will allow Employee to see the Order list.
* The system will allow Employee to see notifications.

# **3.2 Non-functional Requirements**

## **3.2.1 Performance Requirements**

The system should store all the database records of Customers, Employees, and the hire request should be available for use 24\*7 through the server. Also, the application should be user friendly with a proper user interface which makes it easy for the user to understand. All the options should be present in properly accessible places for user convenience.

## **3.2.2 Safety Requirements**

All login ids and passwords of the Admin, Employee and Customer should be protected for privacy using whatever constraints required in the database or the application.

## **3.2.3 Security Requirements**

Passwords of the Admin, Employee and Customer should be protected for privacy using whatever constraints required in the database or the application. User’s password should be saved in encrypted format so that intruder cannot know the password of user. All passwords should be stored as a secure hash of the administrator password.

## **3.2.4 Software Quality Attributes**

## **3.2.4.1 Availability**

The system should run on a variety of operating systems that support the Web browser and has internet connection. The system should run on a variety of hardware.

## **3.2.4.2 Accessibility**

The software will be accessible to admin, police station and police employees.

## **3.2.4.3 Compatibility**

The software will be compatible with multiple platforms.

## **3.2.4.4 Durability**

The software will be tested for working with multiple users and records as system has to manage multiple users (admin, employee) and records (Employee list, Charges, Order List, Notifications).

## **3.2.4.5 Effectiveness**

The software will be made to handle operations effectively.

## **3.2.4.6 Maintainability**

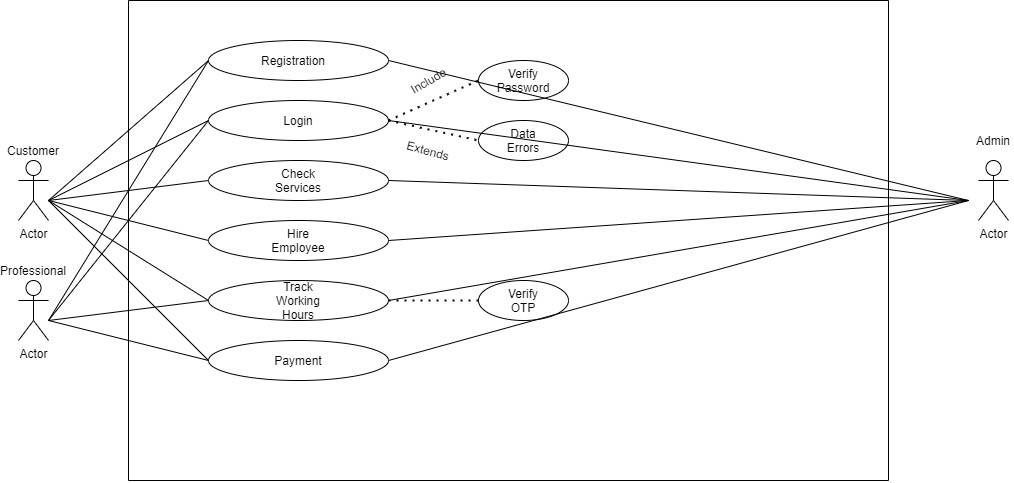
The system should be easy to maintain. There should be a clear separation between the interface and the business logic code. There should be a clear separation between the data access objects that map the database and the business logic code.

**Chapter 4**

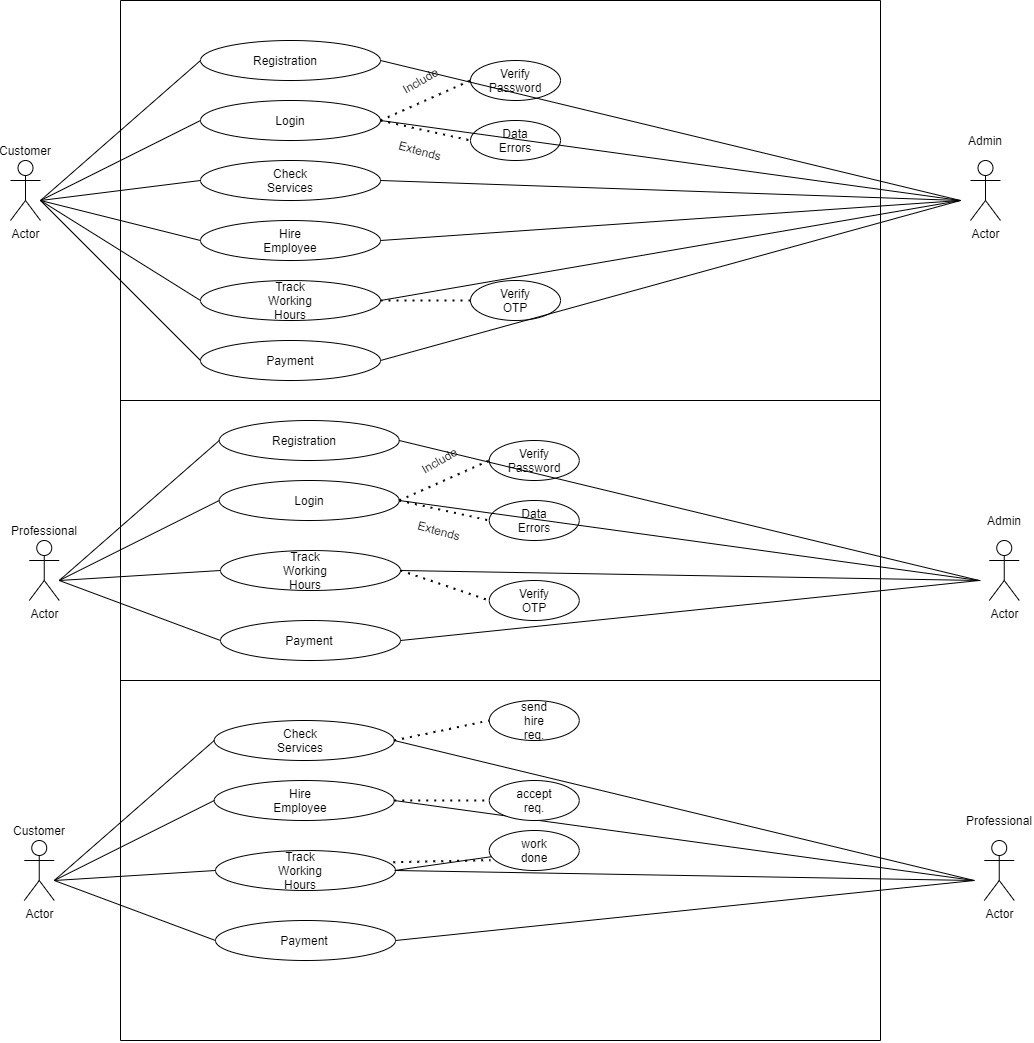
**Analysis Modeling**

## **4.1 Use Case Diagram: -**

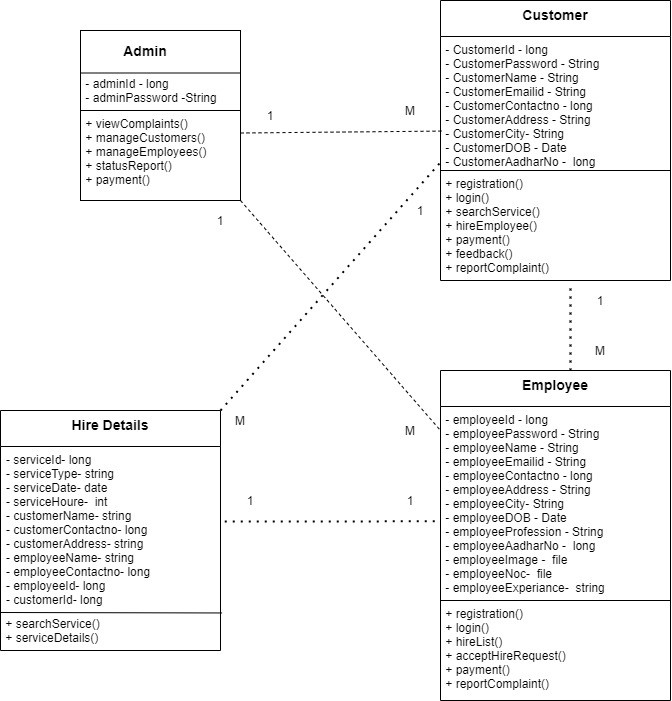
**User Case Diagram: a.1**



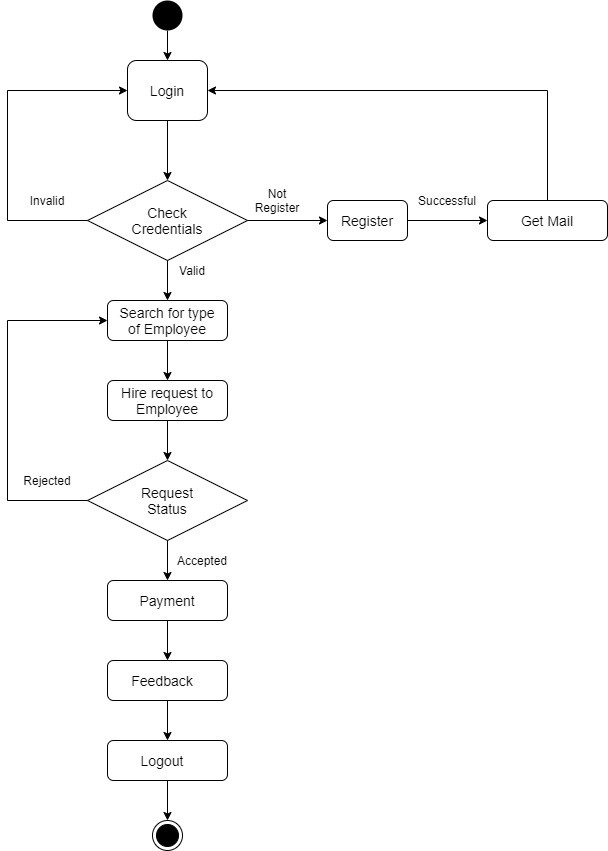
**User Case Diagram: a.2**



4.2 Class Diagram



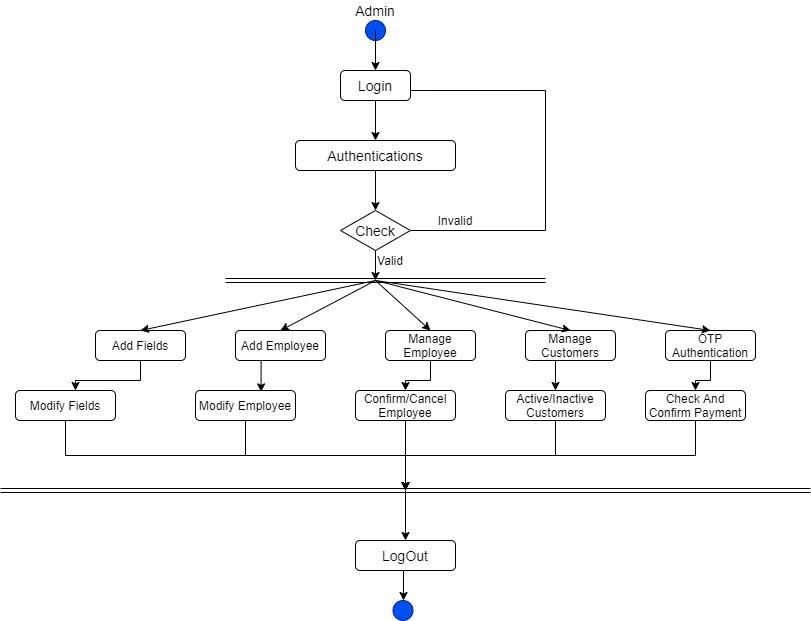
4.3.1 Activity Diagram



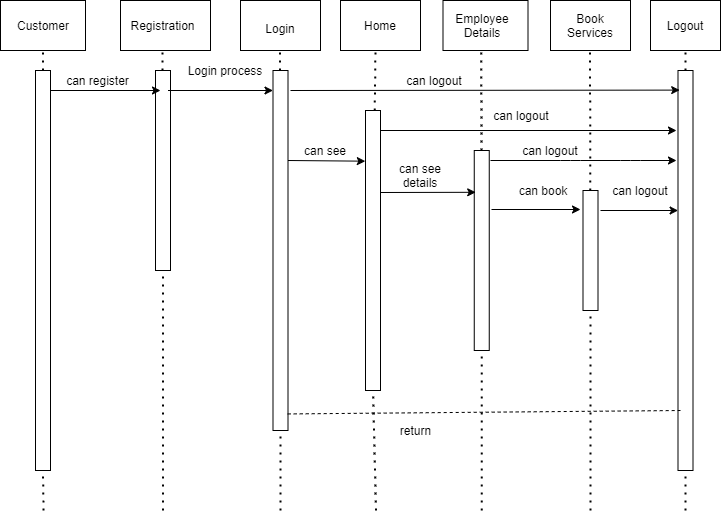
4.3.2 Activity Diagram



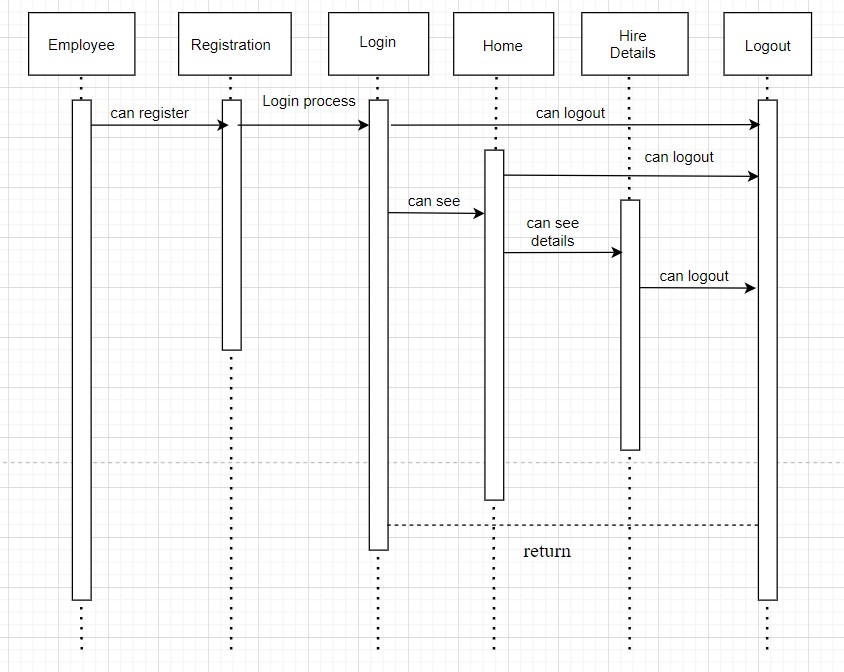
4.3.3 Activity Diagram



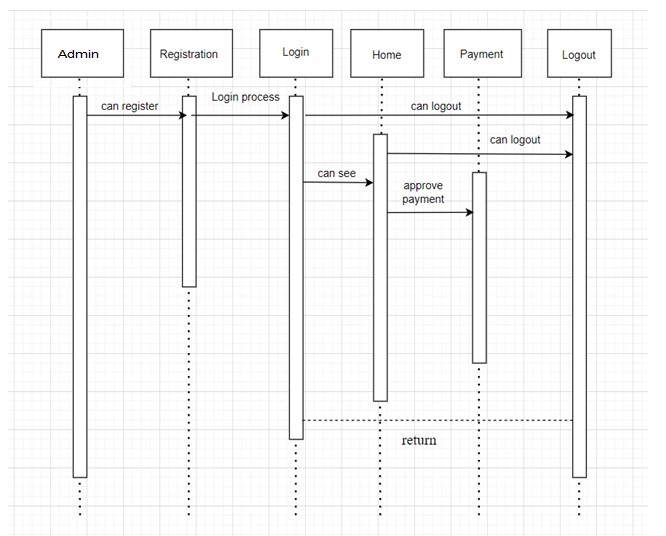
4.4.1 Sequence Diagram



4.4.2 Sequence Diagram



4.4.3 Sequence Diagram

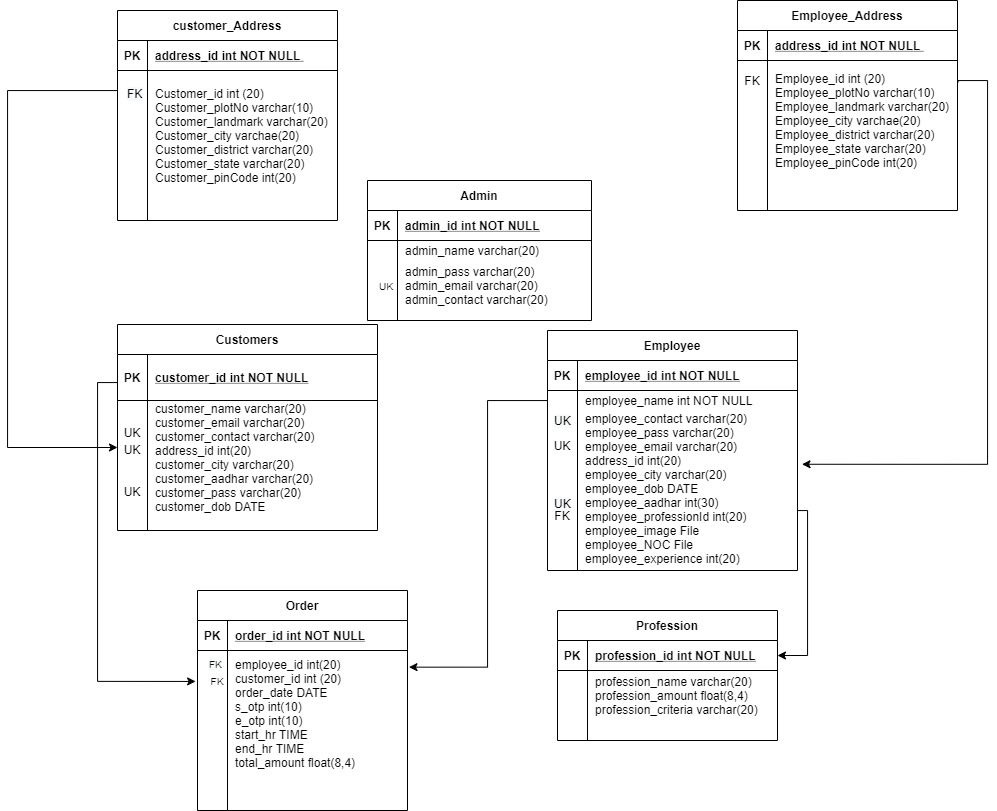


5.0 Gant Chart

**Chapter 5**

**DESIGN**

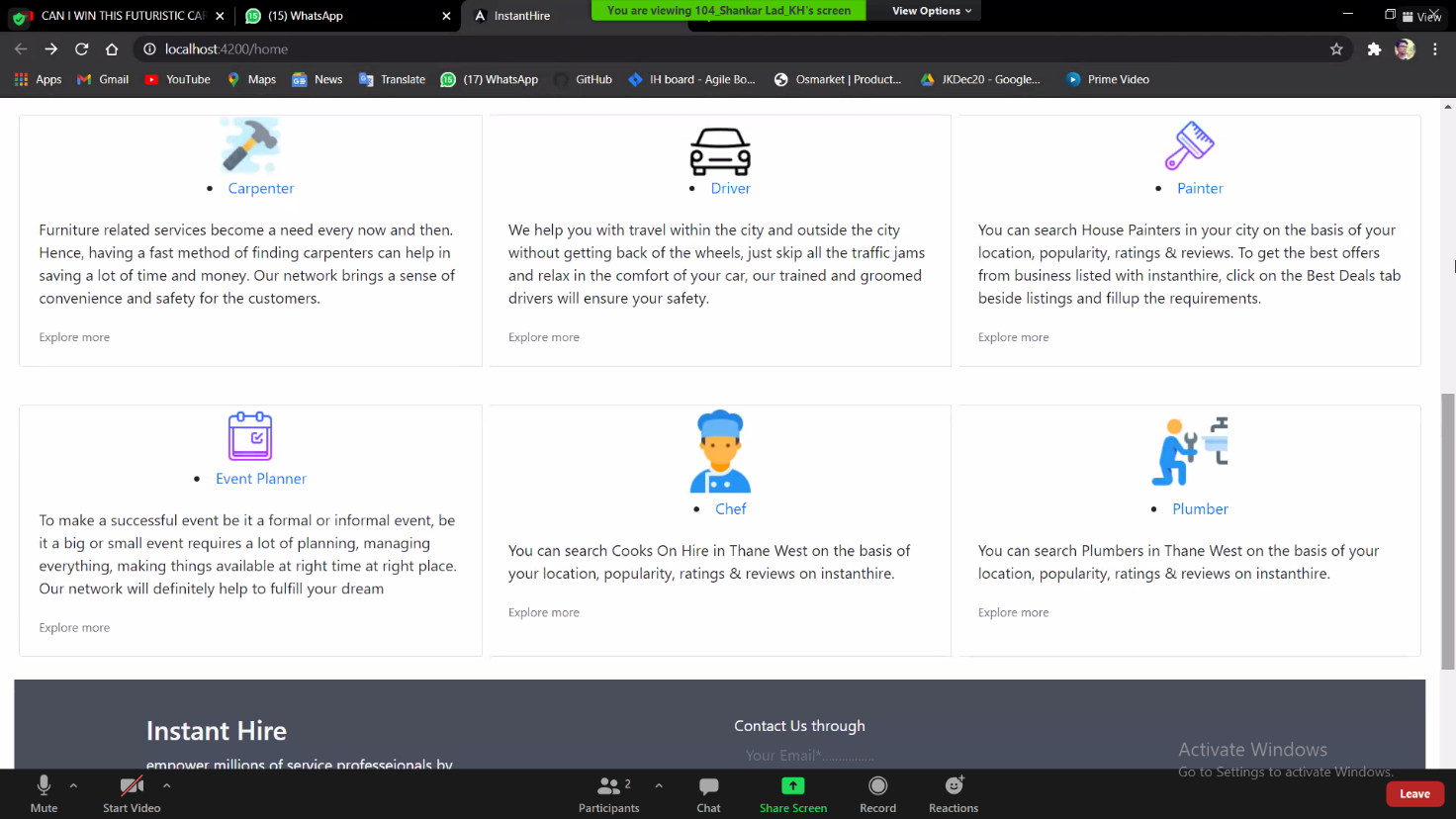
**5.1 Data Modeling *(E-R Model)***

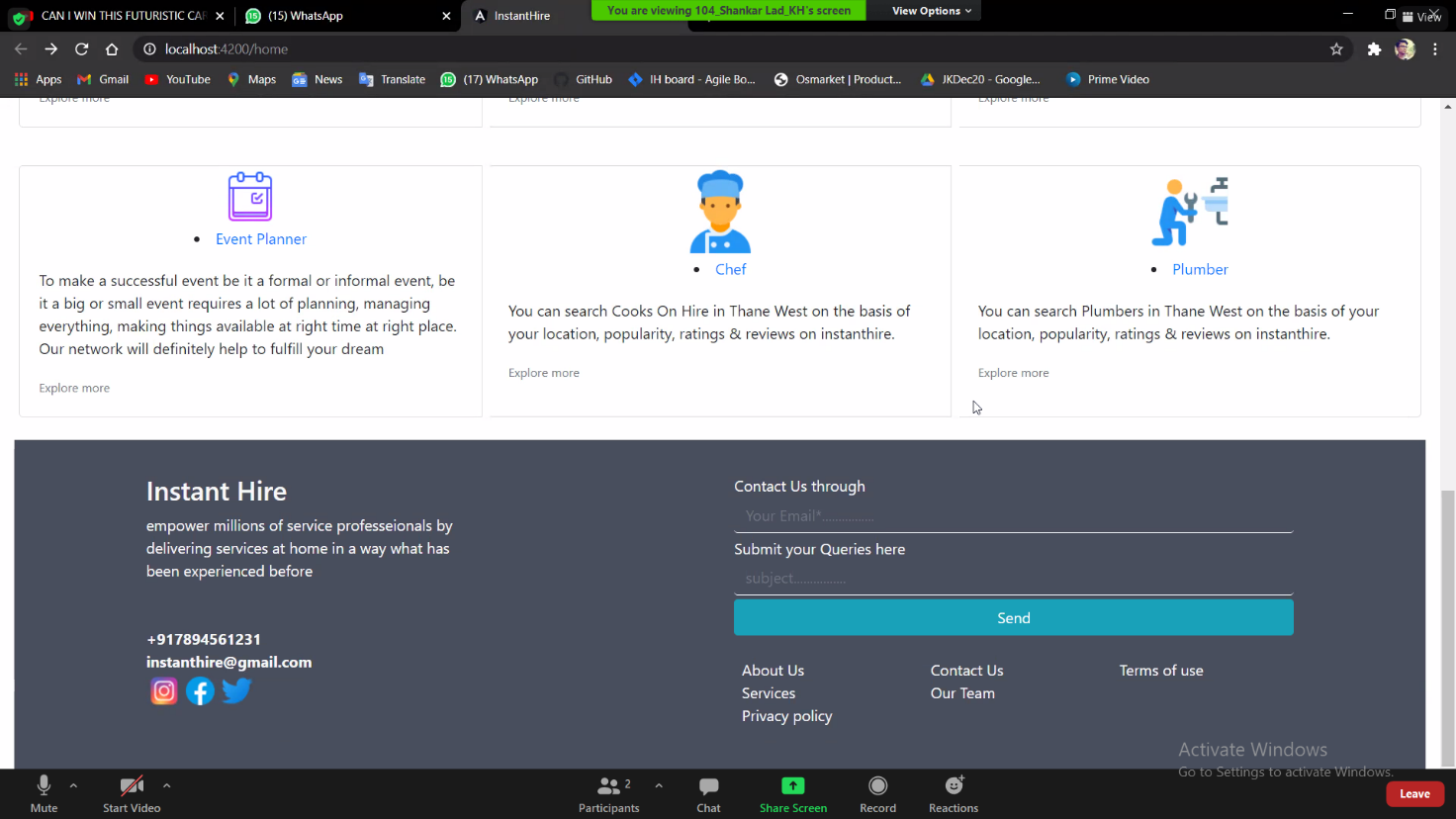


**5.2 User Interface Design** GUI for your project

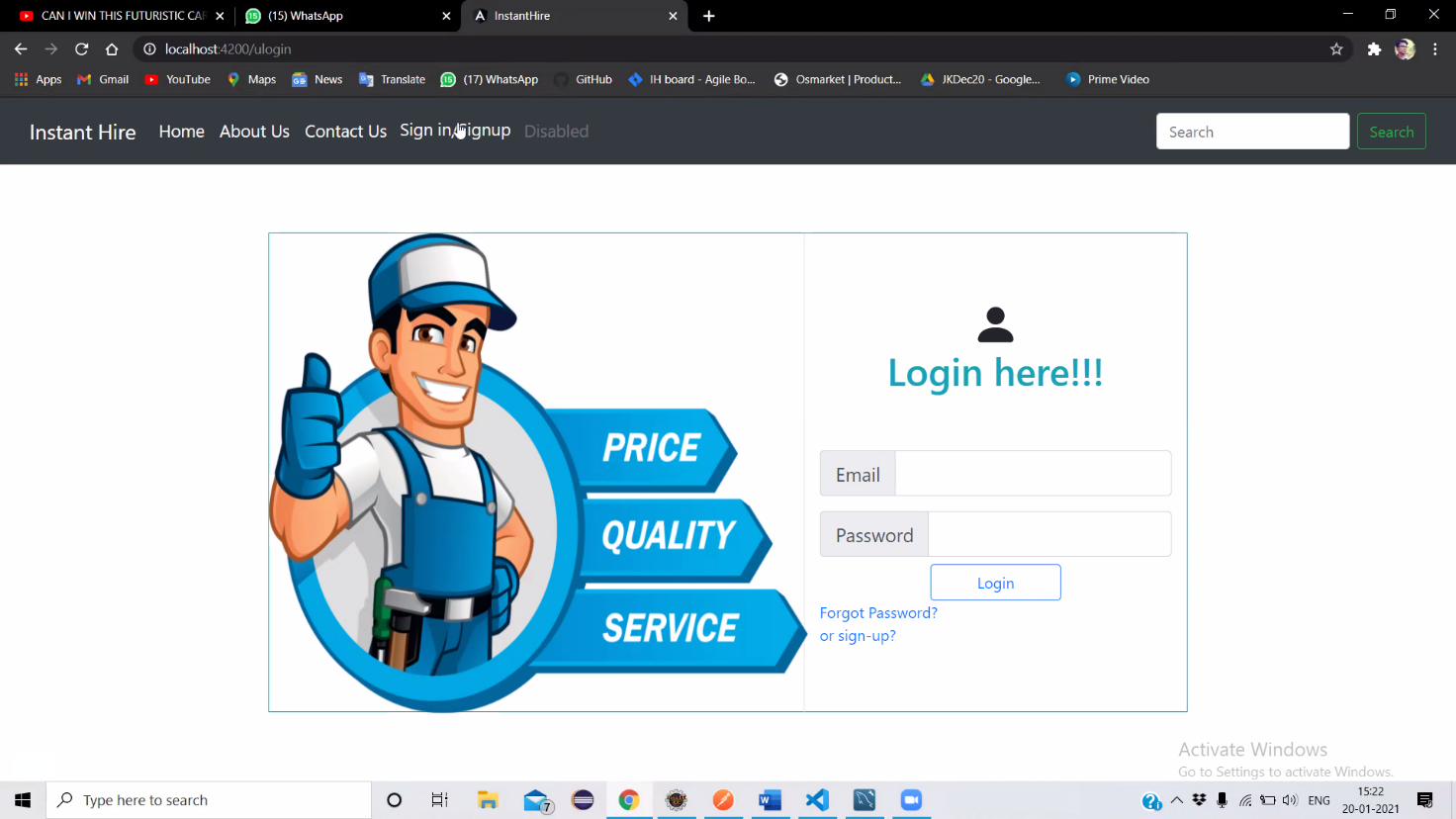
5.2.1 Home Page:

****

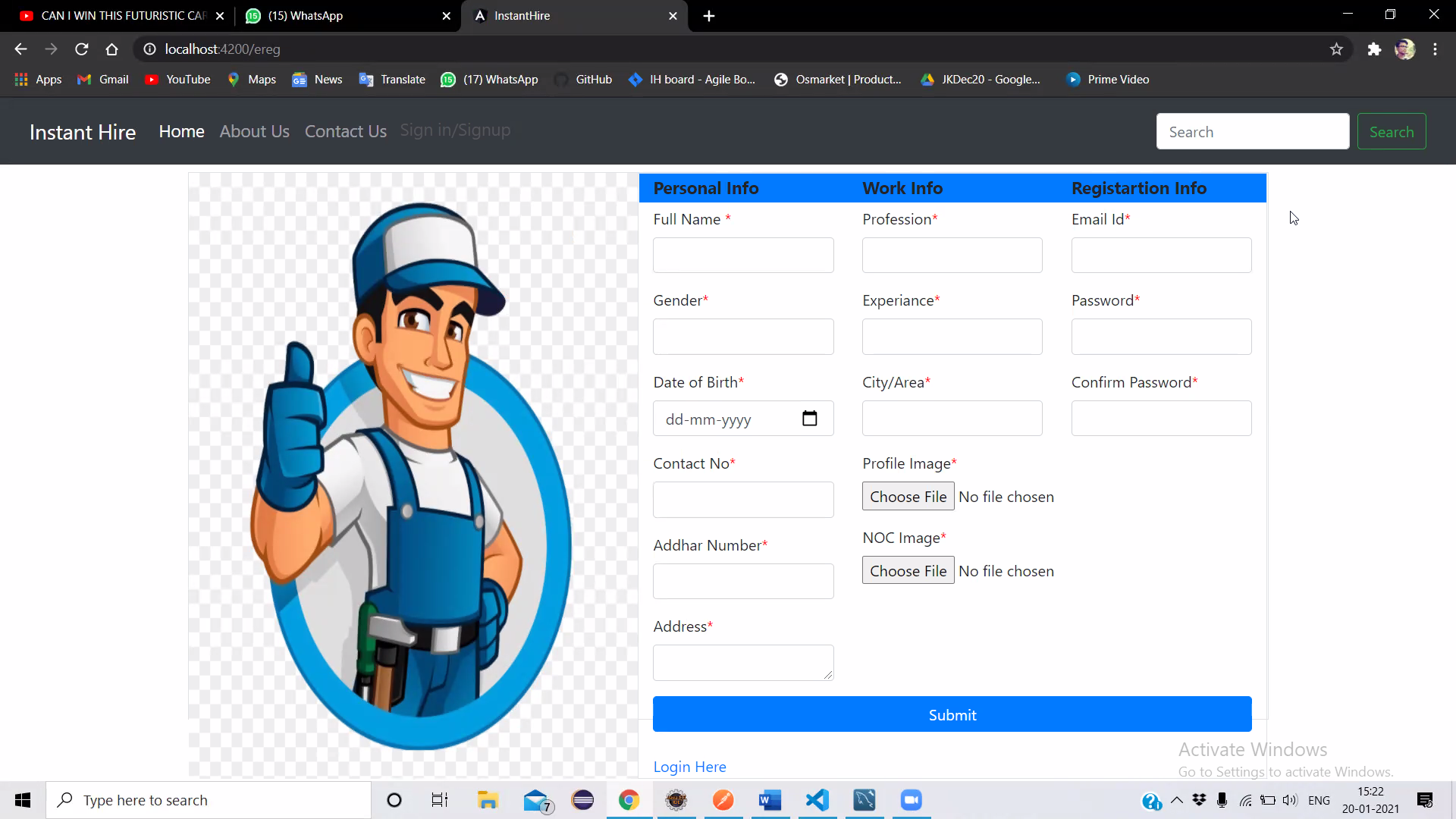




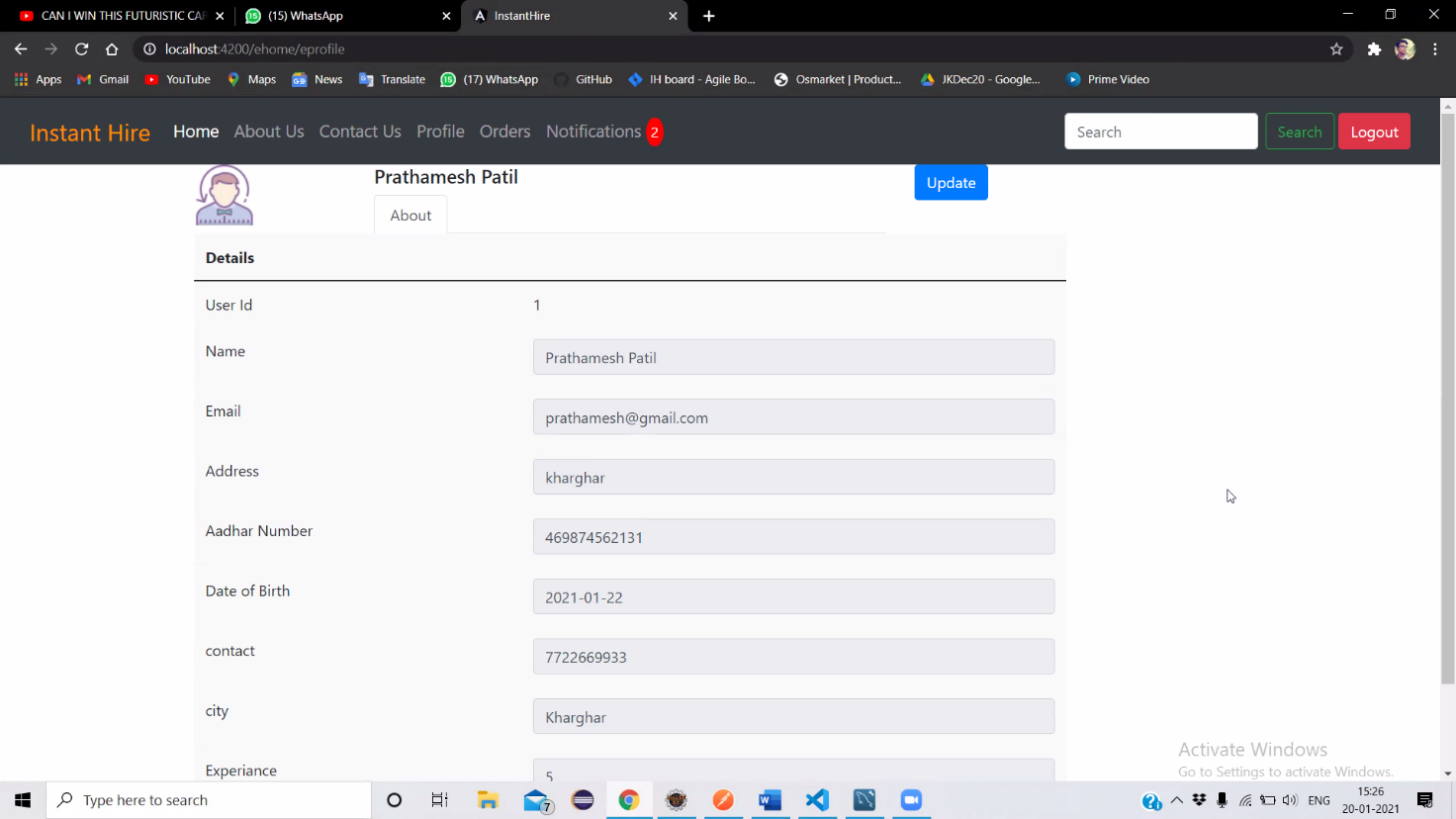
5.2.2 Login Page:



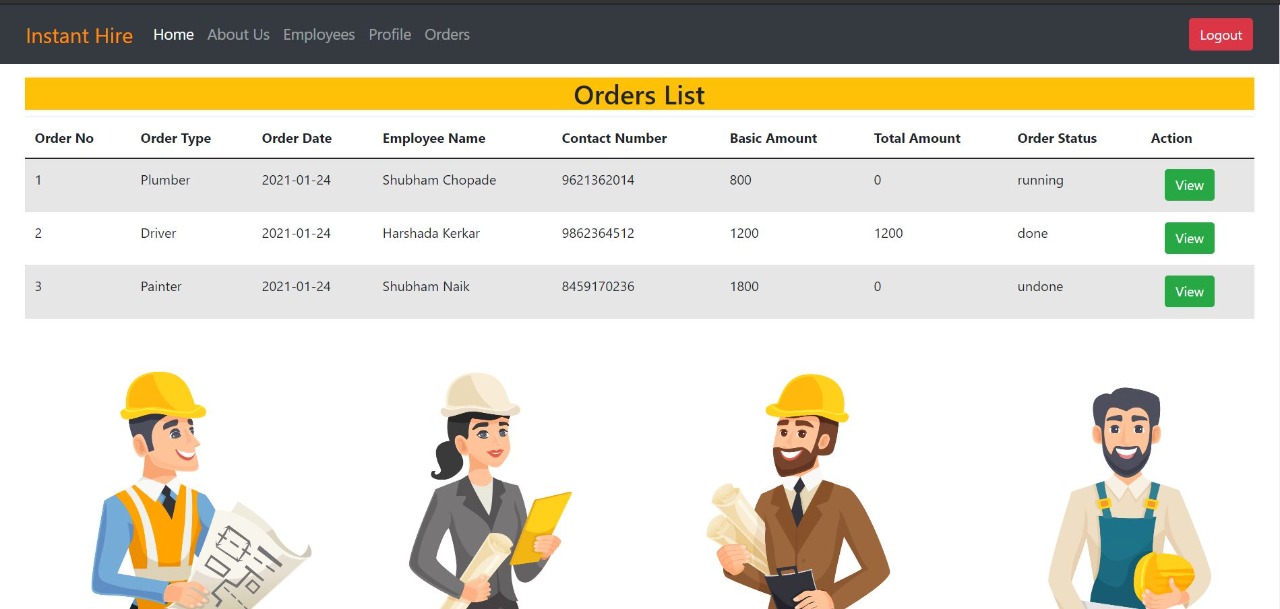
5.2.3 Register Page:



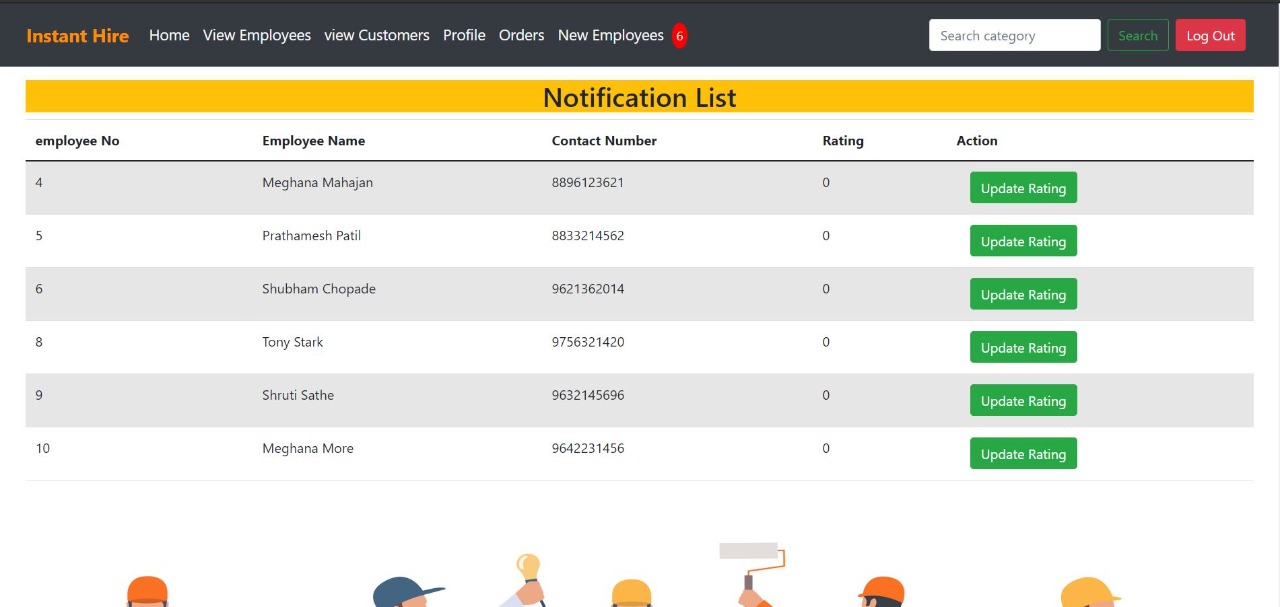
5.2.4 Customer Details Page:



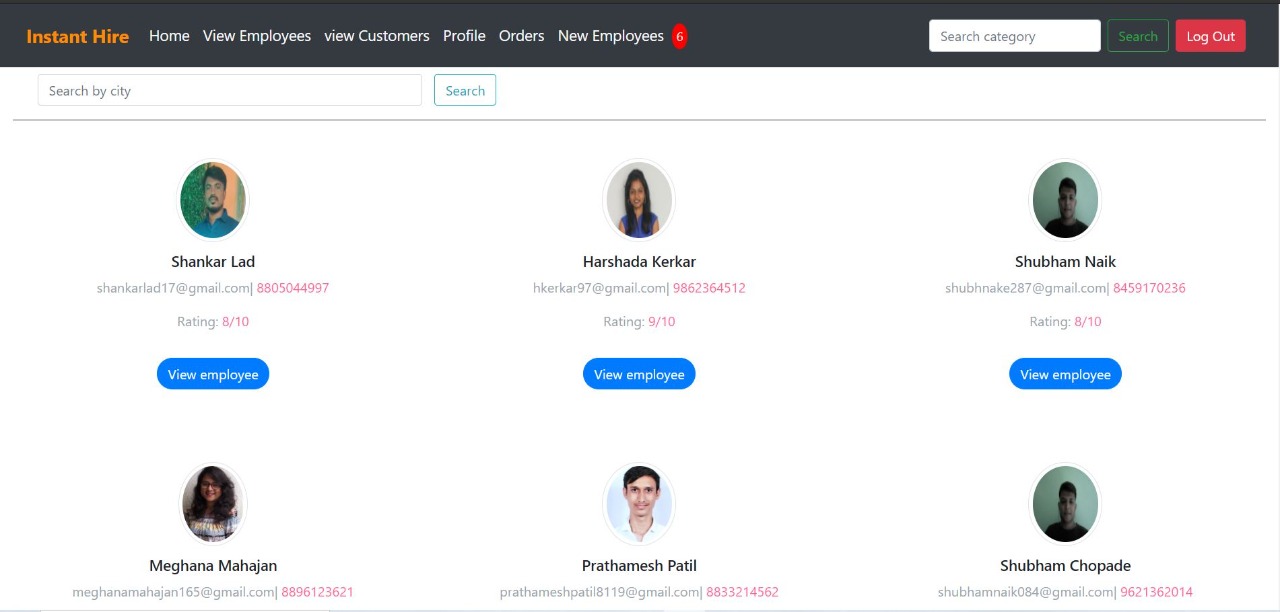
5.2.5 Order List Page:



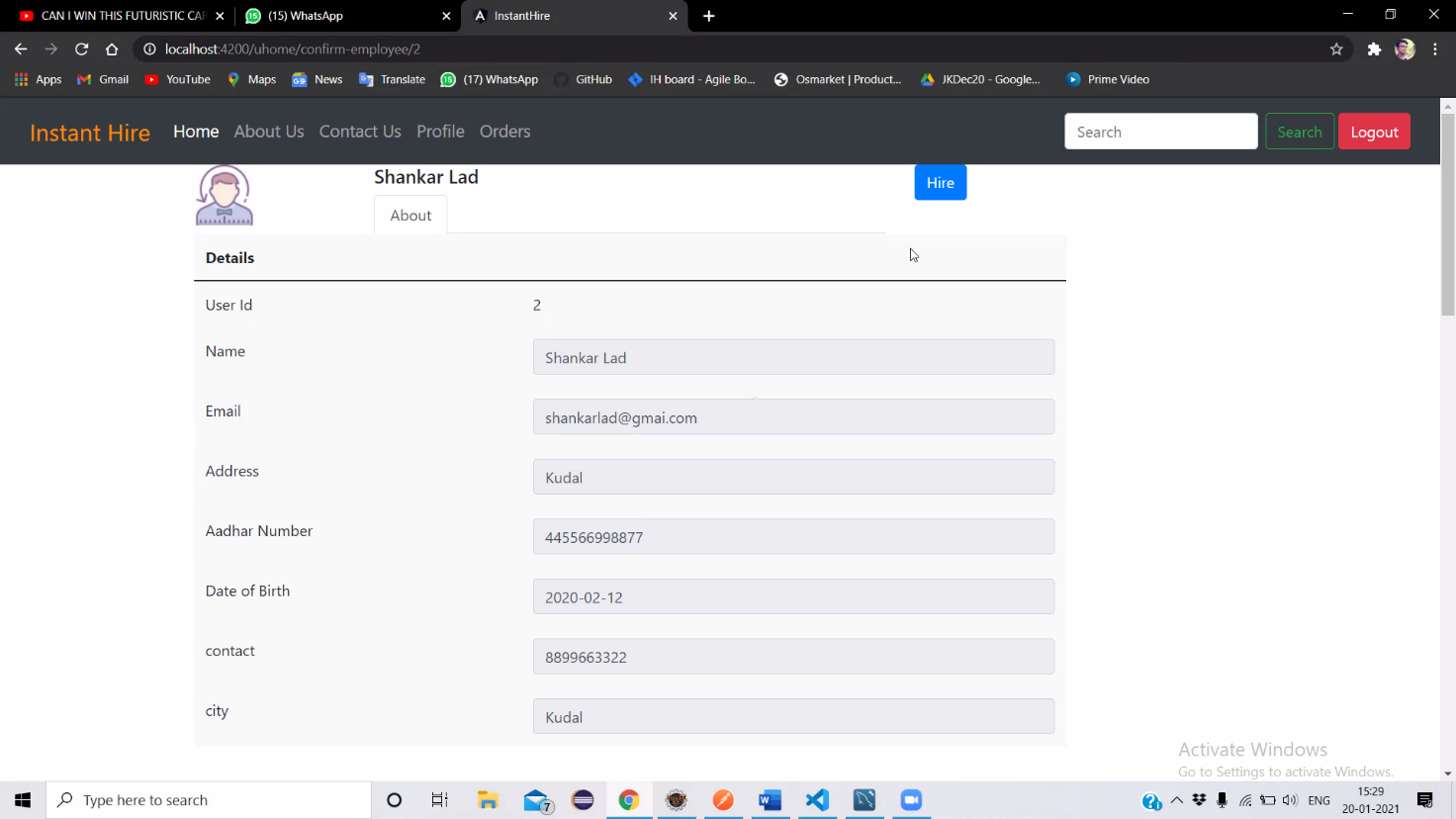
5.2.6 Notification List Page:



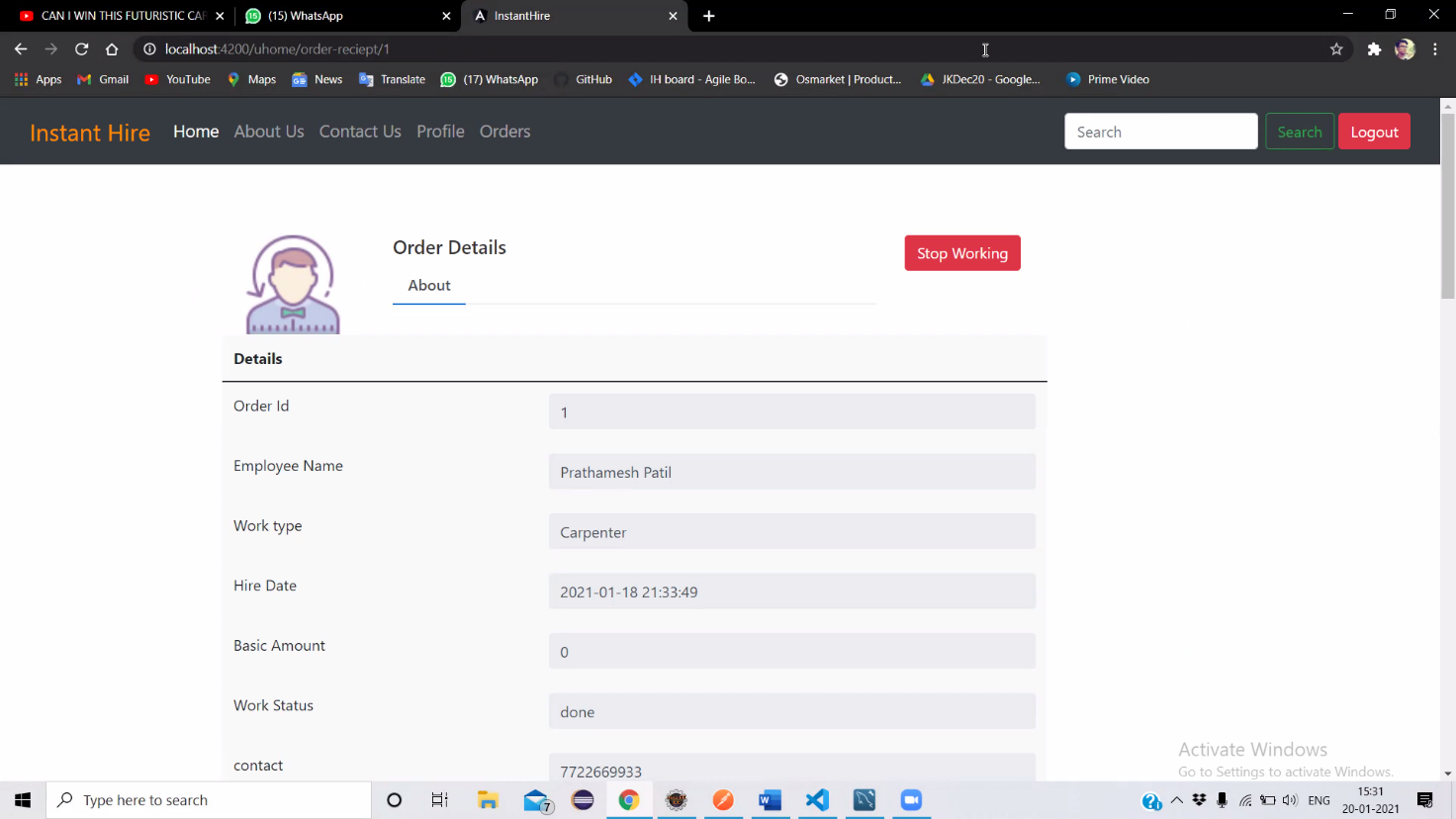
5.2.7 Employee List Page:



5.2.8 Customer Details Page:



5.2.8 Order Details Page:



**Chapter 6**

**IMPLEMENTATION**

**6.1 Create Component:**

@RequestMapping(value="addcust",method=RequestMethod.***POST***)

**public** Customer addCustomer(@RequestBody Customer cust) {

**return** custRepository.save(cust);

}

**6.2 Update Component:**

@CrossOrigin

@PutMapping("/updatecustomer/{id}")

**public** ResponseEntity<Customer> updateCustomer(@PathVariable Long id,@RequestBody Customer customerDetails){

Customer customer = custRepository.findById(id)

.orElseThrow(() -> **new** ResourceNotFoundException("customer Data Not Available for id : " + id));

customer.setCustomerName(customerDetails.getCustomerName());

customer.setCustomerPassword(customerDetails.getCustomerPassword());

customer.setCustomerEmail(customerDetails.getCustomerEmail());

customer.setCustomerContact(customerDetails.getCustomerContact());

customer.setCustomerDOB(customerDetails.getCustomerDOB());

customer.setCustomerAadharNumber(customerDetails.getCustomerAadharNumber());

customer.setCustomerAddress(customerDetails.getCustomerAddress());

customer.setCustomerCity(customerDetails.getCustomerCity());

Customer updatedCustomer = custRepository.save(customer);

**return** ResponseEntity.*ok*(updatedCustomer);

}

**6.3 Delete Component:**

@CrossOrigin

@DeleteMapping("/deletecustomer/{id}")

**public** ResponseEntity<Map<String, Boolean>> deleteCustomer(@PathVariable Long id){

Customer customer = custRepository.findById(id).orElseThrow(() -> **new** ResourceNotFoundException("customer Data Not Available for id : " + id));

custRepository.delete(customer);

Map<String, Boolean> response = **new** HashMap<>();

response.put("Deleted", Boolean.***TRUE***);

**return** ResponseEntity.*ok*(response);

}

**6.4 Image Component:**

@CrossOrigin(origins = "http://localhost:4200")

@PostMapping("/upload/{id}")

**public** BodyBuilder uploadImage(@PathVariable Long id,@RequestParam("imageFile") MultipartFile file) **throws** IOException {

System.***out***.println("emp id : "+id);

System.***out***.println("Original Image Byte Size - " + file.getBytes().length);

ImageModel img = **new** ImageModel(file.getOriginalFilename(), file.getContentType(),

*compressBytes*(file.getBytes()), id);

imageRepository.save(img);

**return** ResponseEntity.*status*(HttpStatus.***OK***);

}

@CrossOrigin(origins = "http://localhost:4200")

@GetMapping(path = { "/show/{id}" })

**public** ImageModel getImage(@PathVariable("id") **long** imgid) **throws** IOException {

**final** Optional<ImageModel> retrievedImage = imageRepository.findByEmployeeId(imgid);

ImageModel img = **new** ImageModel(retrievedImage.get().getName(), retrievedImage.get().getType(),

*decompressBytes*(retrievedImage.get().getPicByte()),imgid);

**return** img;

}

// compress the image bytes before storing it in the database

**public** **static** **byte**[] compressBytes(**byte**[] data) {

Deflater deflater = **new** Deflater();

deflater.setInput(data);

deflater.finish();

ByteArrayOutputStream outputStream = **new** ByteArrayOutputStream(data.length);

**byte**[] buffer = **new** **byte**[1024];

**while** (!deflater.finished()) {

**int** count = deflater.deflate(buffer);

outputStream.write(buffer, 0, count);

}

**try** {

outputStream.close();

} **catch** (IOException e) {

}

System.***out***.println("Compressed Image Byte Size - " + outputStream.toByteArray().length);

**return** outputStream.toByteArray();

}

// uncompress the image bytes before returning it to the angular application

**public** **static** **byte**[] decompressBytes(**byte**[] data) {

Inflater inflater = **new** Inflater();

inflater.setInput(data);

ByteArrayOutputStream outputStream = **new** ByteArrayOutputStream(data.length);

**byte**[] buffer = **new** **byte**[1024];

**try** {

**while** (!inflater.finished()) {

**int** count = inflater.inflate(buffer);

outputStream.write(buffer, 0, count);

}

outputStream.close();

} **catch** (IOException ioe) {

} **catch** (DataFormatException e) {

}

**return** outputStream.toByteArray();

}

**6.5 Send Email :**

@CrossOrigin

@GetMapping("/getcustbyemail/{emailId}")

**public** Customer getCustomerByEmail(@PathVariable String emailId){

Customer cust=custRepository.findByCustomerEmail(emailId);

**if**(Objects.*nonNull*(cust)) {

System.***out***.println(cust);

**return** cust;

}**else** {

System.***out***.println("customer not found");

**return** **null**;

}

}

**6.5 Get List of Employees :**

@CrossOrigin

@GetMapping("/custlist")

**public** List<Customer> getCustomerList(){

**return** custRepository.findAll();

}

**Chapter 7**

**TESTING**

**7.1 Test Cases**

**7.1.1 Introduction**

* The aim of testing process is to identify all defects in a software product. Testing is any activity aimed at evaluating the software for quality results is produces and the quality of results it can handle. Testing is an operation to detect the differences between the expected result and the actual result.
* Our goal is to design a series of test cases that would have a high likelihood of finding errors. The software testing technique provides systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domain of the program to uncover errors in program function, behavior and performance.

**7.1.2 System test objective and scope**

* **Software tested from two different perspective**

1. Internal program logic is exercised using “White Box” test cases design technique.
2. In this technique internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security.
3. Software requirements are exercised using “Black Box” test case design techniques, involves testing from an external or end-user type perspective.

* **The main aim to test this is to insure that**

1. The proposed system permits only secure and authenticate access.
2. Thus requires user to enter URL in correct format.
3. Does all client side as well as server side validation time to time as per the need.
4. Appropriate alerts are generated as per the condition for user convenience.
5. Database is updated time to time as the user transaction process proceeds.

* **Tested Items**

1. All functionality of Admin Domain
2. All functionality of Customer Domain
3. All functionality of Employee Domain

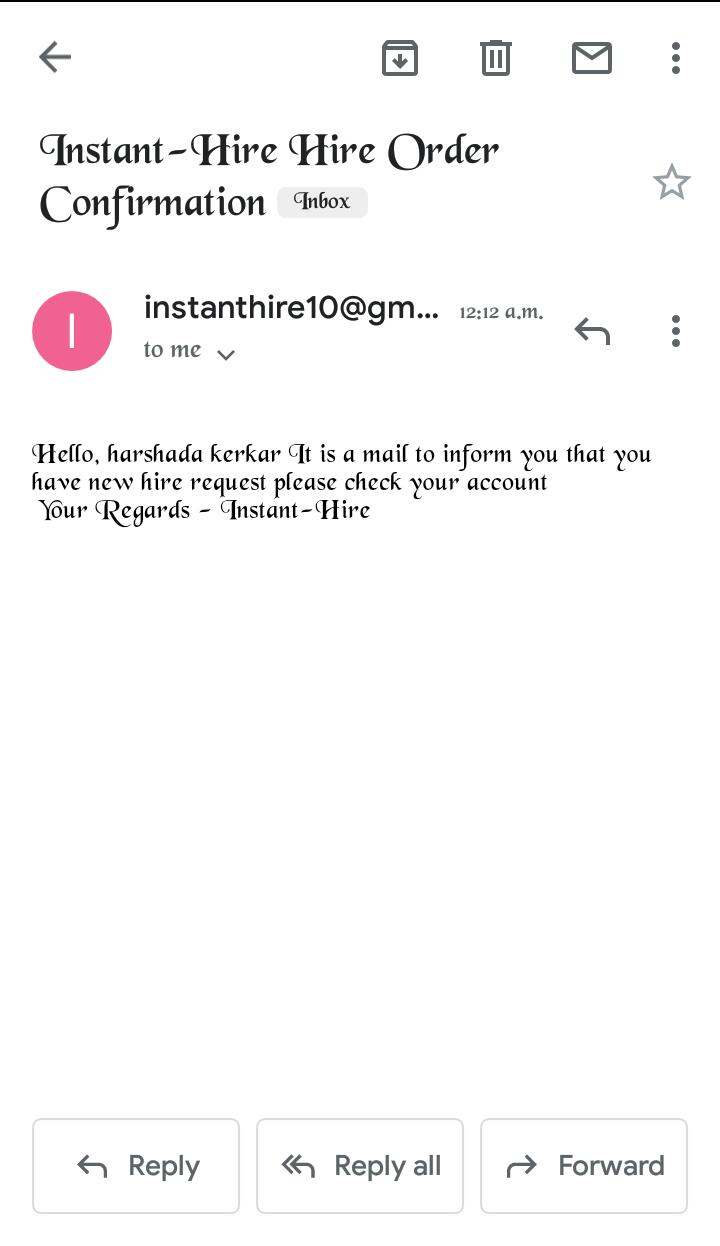
**7.2 TEST CASES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Id** | **Items to be tested** | **Steps** | **Input** | **Actual Output** | **Expected**  **Output** | **Pass/Fail** |
| 1. | Customer Login  Employee Login | Enters email & password | Email & password  Server side validations | Redirect to home page | Display homepage Successful | Pass |
| 2. | Admin Login | Enter email & OTP(get by mail) | Email  & OTP | Redirect to home page | Display homepage Successful | Pass |
| 3. | Customer Registration  Employee Registration | Enters required form data  & check validations | Enters  Data  & check for client side validations | Message successful login | Redirect to Login page | Pass |
| 4. | Customer search for Employee | Click on category or search by city | Click on category  Or input in search bar | Show searched category list | Display card type list | Pass |
| 5. | Customer Hire Employee | Click on Hire button | Click button | Message order placed | Display place order with undone status | Pass |
| 6. | Start Work | Click start button | Click start | Order status changed to running | Display place order with running status | Pass |
| 7. | Stop Work | Click stop button | Click stop | Order status changed to done | Display place order with done status | Pass |
| 8. | Employee  Notification | Load home page | Login & load home page | Show new order notification | Display on header total new orders | Pass |
| 9. | Admin Notification | Load Home | Login & load home | Show new register Employees | Display on header list of new register employee | Pass |
| 10. | Update Profile  Customer, Employee | Load update from & change inputs | Enter updated Input | Show profile with updated inputs | Display profile with updated inputs | Pass |
| 11. | Admin Login | Enter Mail & login through received OTP on mail | Enter mail & click send OTP | OTP received on register valid Mail | Enter OTP & if it matched redirect to Homepage | Pass |
| 12. | Forgot password | Enter mail & click send OTP | Enter mail & click send OTP | Old Password send through mail | Input Old password & new Password & update | Pass |

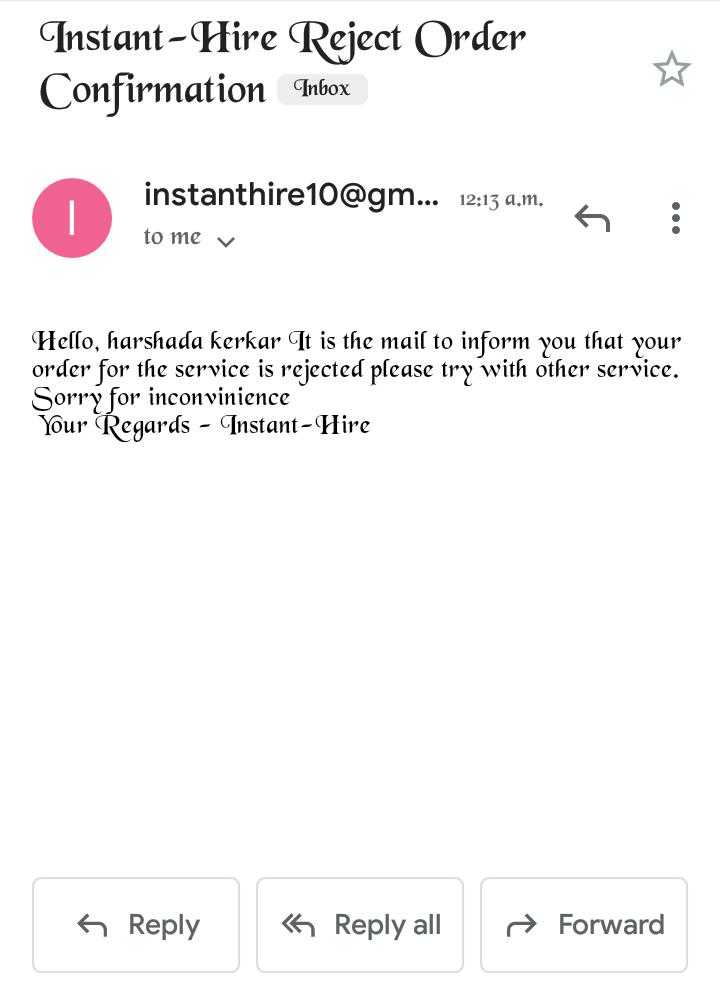
**Chapter 8**

**Results and Discussions**

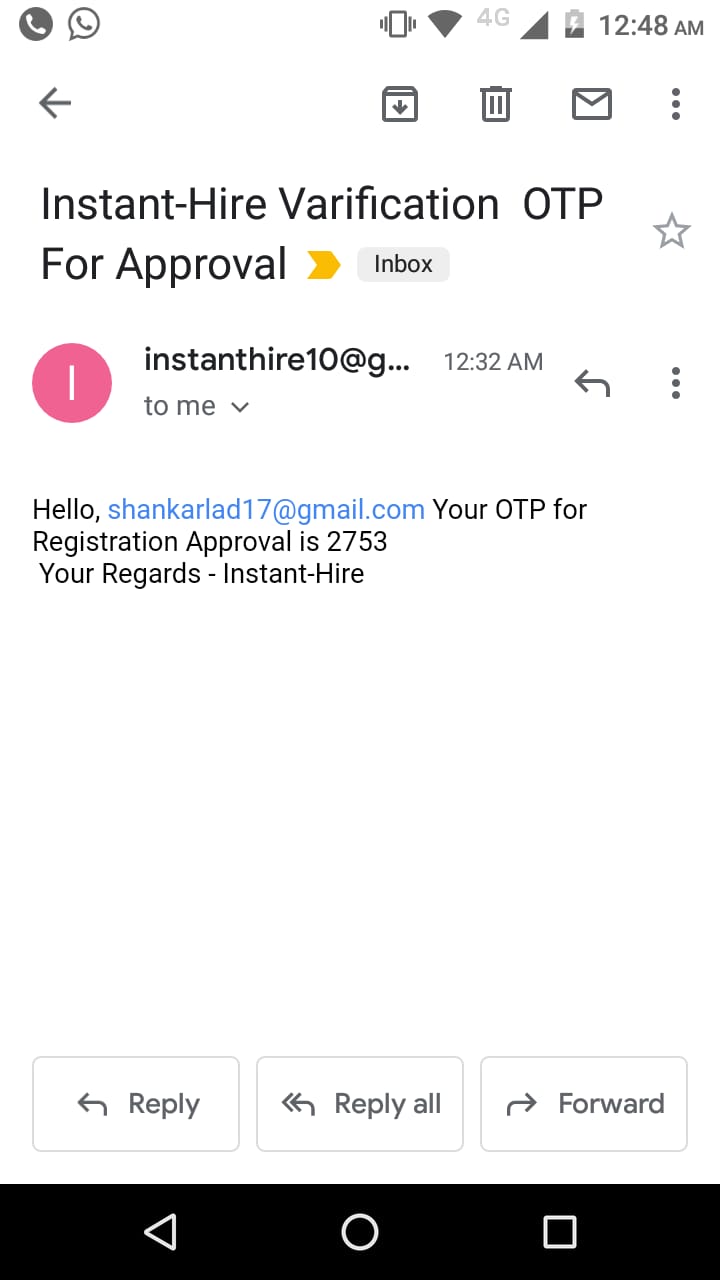
8.1 Reject Page:



8.2 Hire Page:



8.3 Admin Login OTP Page:



**Chapter 9**

**Conclusions**

* The company should take proper feedback from customer after providing services which also shows professionalism….
* The hygiene level of the service provider is also very important for the customer; the service provider should be hygiene & should be very professional in the terms of services…and all this things we are added in our site.
* Instant-Hire somehow help to solve the problem of unemployment. And also it helps the people who lost their job because of covid-19.
* The Instant-Hire representative should have the complete & proper information about the services and products which he/she is giving to customer.
* And at last but not the list as we planned we implemented all the things and try to solve the problem and somehow we succeed in that…

**Future Scope**

* As we see that Instant-hire is very useful for all the people across India. The Platform Helps customer’s book reliable home services Like Cleaning, plumbing, carpentry, painting etc... And also because of Covid-19 Pandemic-Instant hire can serve more people across India…
* We can implement it as website or Platform like an Urban clap….
* Now a day it’s very hard to search trustworthy employee and searching job our project can easily provide services to users and job to jobless people.
* We can also use our website as a consultant website which can provide lots of jobs to the many people across India…
* So Our Website Mission is to empower of service professionals by delivering services at-home in a way that has never been experienced before and we are working on it though it…..

**Literature Cited**

**IEEE standard**

**Book,**

[1] .N. M. Indravasan, Adarsh G, Shruthi C, Shanthi K, “An Online System for Household Services” International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, May 2018

[2]. Sheetal Bandekar, Avril D’Silva, “Domestic Android

Application for Home Services” International Journal of Computer Applications, ISSN No.0975 – 8887, Volume 148 – No.6, August 2016.

[3]. Shahrzad Shahriari, Mohammadreza Shahriari, Saeid gheiji. “ Ecommerce And It Impaction Global Trend And Market” International Journal of Research – Granthaalayah. Vol.3 (Iss.4): April 2015.

[4]. Bo Zhang, Ruihan Yong, Meizi Li, Jianguo Pan, Jifeng Huanglaa,“ A Hybrid Trust Evaluation Framework for E-commerce in Online Social Network: ” 2169-3536, IEEE. Translations and content mining are permitted for academic research, 2016.

[5]. Chenggang Zhen, Peng Cheng. “Construction of campus trading platform based on third-party online payment” 2nd International Conference on Industrial and Information Systems, IEEE, 2010.

[6]Sujit Kumar Basak,Irene Govender.“Examining the Impact of Privacy, Security, and Trust on the TAM and TTF Models for Ecommerce Consumers: A Pilot Study”, IEEE, 2009.

**Acknowledgements**

We express our deep sense if gratitude and sincere regards to our project guides Poorti Sharma , Aishwarya Kale and faculty Supervisor Sohan More **,** for there valuable supervision, cooperation and devotion of time that has given to our project.

We are also grateful to Director **Dr. C. P. Johnson,** for his facilities extended during project work and his personal interest and inspiration.

We wish to express our profound thanks to **Mrs. Elizabeth Jorge** for providing necessary facilities to make this project success.

Finally we would like to thank all those who directly or indirectly helped us during the work. We also owe our sincere thanks to all faculty members of DAC department who have always extended a helping hand.

1. Harshada Kerkar

2.Shankar Lad

3.Prathamesh Patil

4.Meghana Mahajan

5. Shubham Naik