Mini Project 4

ITMD 515

Git Url : <https://bitbucket.org/pnaikaitmd451515s/pnaika-mp4>

**TABLE OF CONTENT :**

[Objective 3](#_Toc416872867)

[About the application: 3](#_Toc416872868)

[Requirement fulfillment: 3](#_Toc416872869)

[Design : 3](#_Toc416872870)

[Domain Model: 3](#_Toc416872871)

[Business Layer: 4](#_Toc416872872)

[Presentation Layer: 4](#_Toc416872873)

[Development Insights: 4](#_Toc416872874)

[Installation Information: 4](#_Toc416872875)

[Software required to install, 4](#_Toc416872876)

[Steps involve in Compilation and Run: 4](#_Toc416872877)

[Glassfish security realm: 4](#_Toc416872878)

[Security Implementations: 5](#_Toc416872879)

[Screen Shots: 6](#_Toc416872880)

[Java Doc 15](#_Toc416872881)

[CRUD method in EJB 15](#_Toc416872882)

[Expected Results: 16](#_Toc416872883)

## 

## Objective

The purpose of this assignment is to explore Java EE service layer components of an enterprise application, integrating JPA domain model with the service layer, and introducing security to the application.

## About the application:

Global Parking System is a parking application which keeps information about the vehicle, parking details, Billing information that are parked in XYZ Company’s paid parking arena. In this paid parking arena, customers are supposed to pay for parking their vehicle on hourly manner.

This application will have two actors, employee and customer. Where each actor will have predefined role and functionality

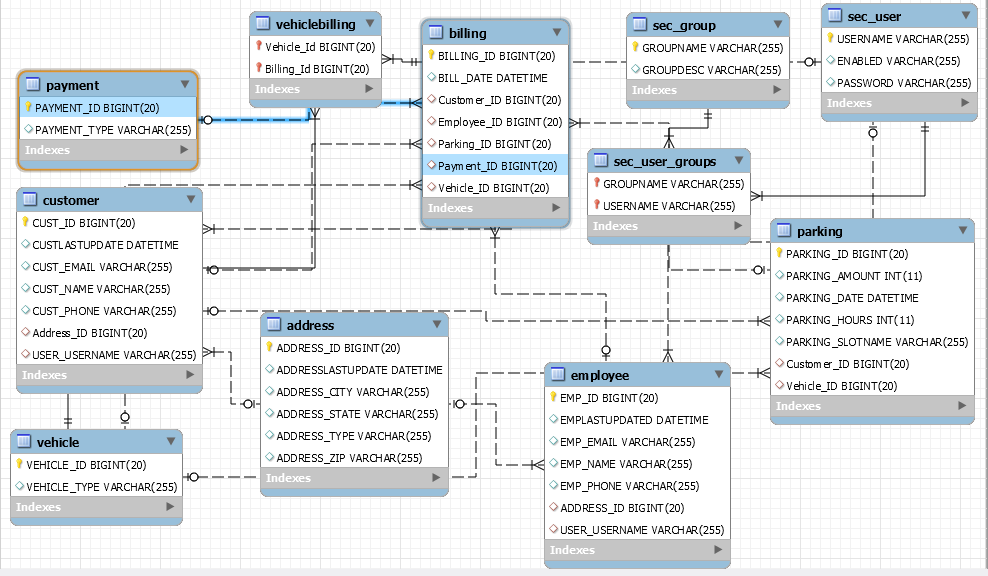
In this assignment to showcase security implementation there are different behavior for each role, I.e, Customer can view only his information alone and employee and see all the customer’s and employee details.

## Requirement fulfillment:

1. Included all appropriate dependencies in pom.xml
2. Created pool and resource to make jdbc connection with naming convention as specified in the requirements, Also there is persistence unit.
3. Each entity class has its own corresponding EJB Stateless component which implements all the CRUD operation on the database and business methods (which interact with the view/ presentation layer).
4. All entity classes extend an abstract class to implement certain operations that are common to all the entities.
5. The data is persisted into the database using a singleton EJB annotated with @Startup.
6. Passwords have been hashed using MD5 hashing approach, Introduced security to the application that incorporates both authentication and authorization using Java EE security.
7. Customer and Employee are two roles in the application. Customers will be able to see their information alone whereas Employees are will be able to see all the customer and employee details

## Design :

### ER Diagram :



### Domain Model:

1. The domain model design is exactly that of MP3.
2. Pojo classes have been created and appropriate annotations have been added to create the table schema and relationships between different tables.
3. Separate Entity class have been created for application security (User and Group).

### Business Layer:

1. Each entity class has its own corresponding EJB Stateless component which implements all the CRUD operation on the database and business methods.
2. All entity classes extend an abstract class to implement certain operations that are common to all the entities.
3. The data is persisted into the database using a singleton EJB annotated with @Startup.
4. Passwords have been hashed using MD5 hashing approach.

### Presentation Layer:

1. Four html files are used to display the Main page, Log in page, error page and 403 handled error page.
2. An index.html and login.html handle the application entry point view.
3. A servlet has been created to handle application which navigates to customer portal, employee portal logout.

## Development Insights:

* Experience includes exploring more in JPA, Security related coding and implementation of domain model
* Good handson on JPA
* Got familiar with Persistence, Implementation of relationships in JPA and applying security to the application
* JPQL has been used to write queries

## Installation Information:

### Software required to install,

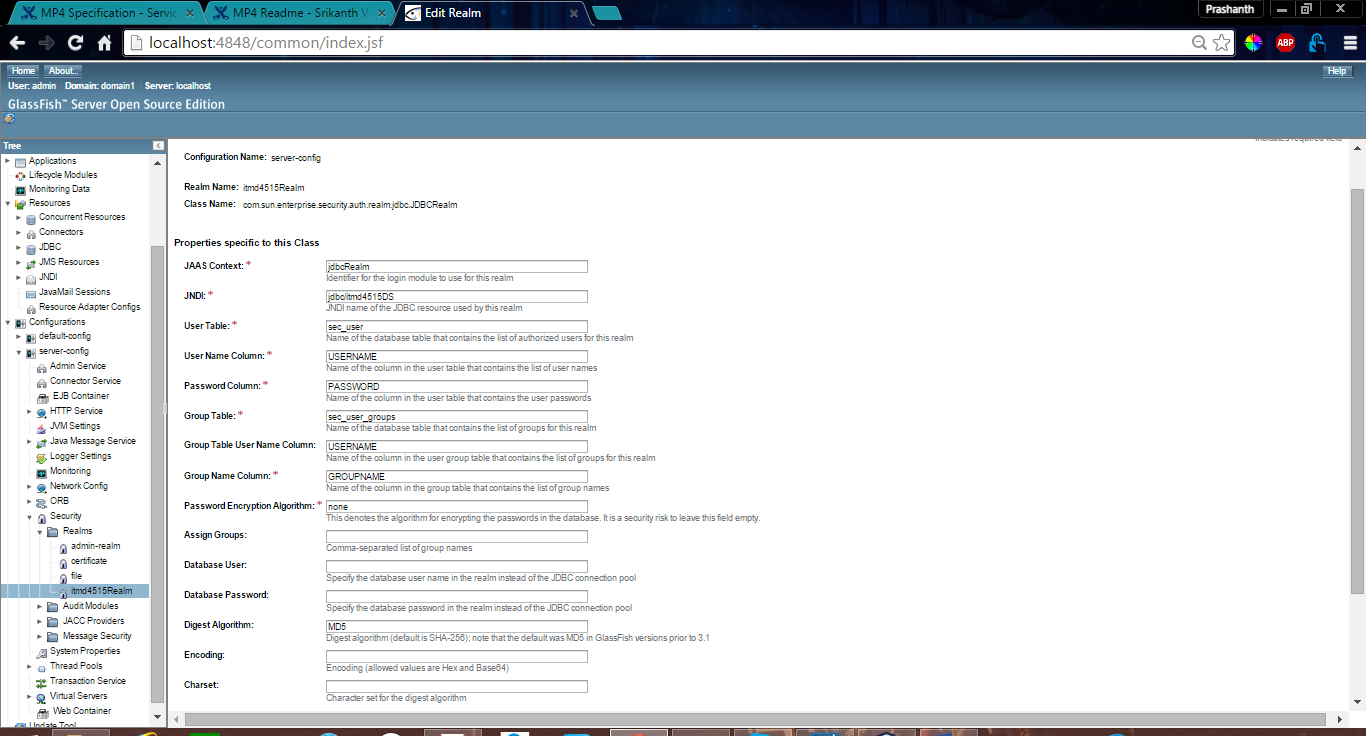
1. NetBean 8
2. MySQL
3. Latest JDK 8

### Steps involve in Compilation and Run:

1. Create Java classes with entities in it, Define attribute type and relationships
2. Create Persistence Unit.
3. Create web.xml and sun-web.xml to define security
4. Code html pages for login and to handle errors, Create Servlets for the transactions.
5. Build and Run the code

### Glassfish security realm:

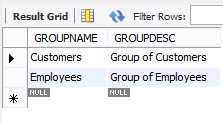
1. Navigate to the Services Tab in NetBeans and expand Servers.
2. Select GlassFish Server 4 and right click to select “Domain Admin Console”.
3. The Domain Admin Console for GlassFish server will open up in a web browser.
4. Navigate to Configurations -> server-config -> Security -> Realms.
5. Create a new Realm itmd4515Realm as show below.



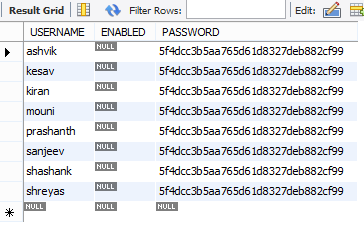
## 

## Security Implementations:

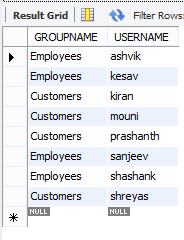
Group Table :



User Table :

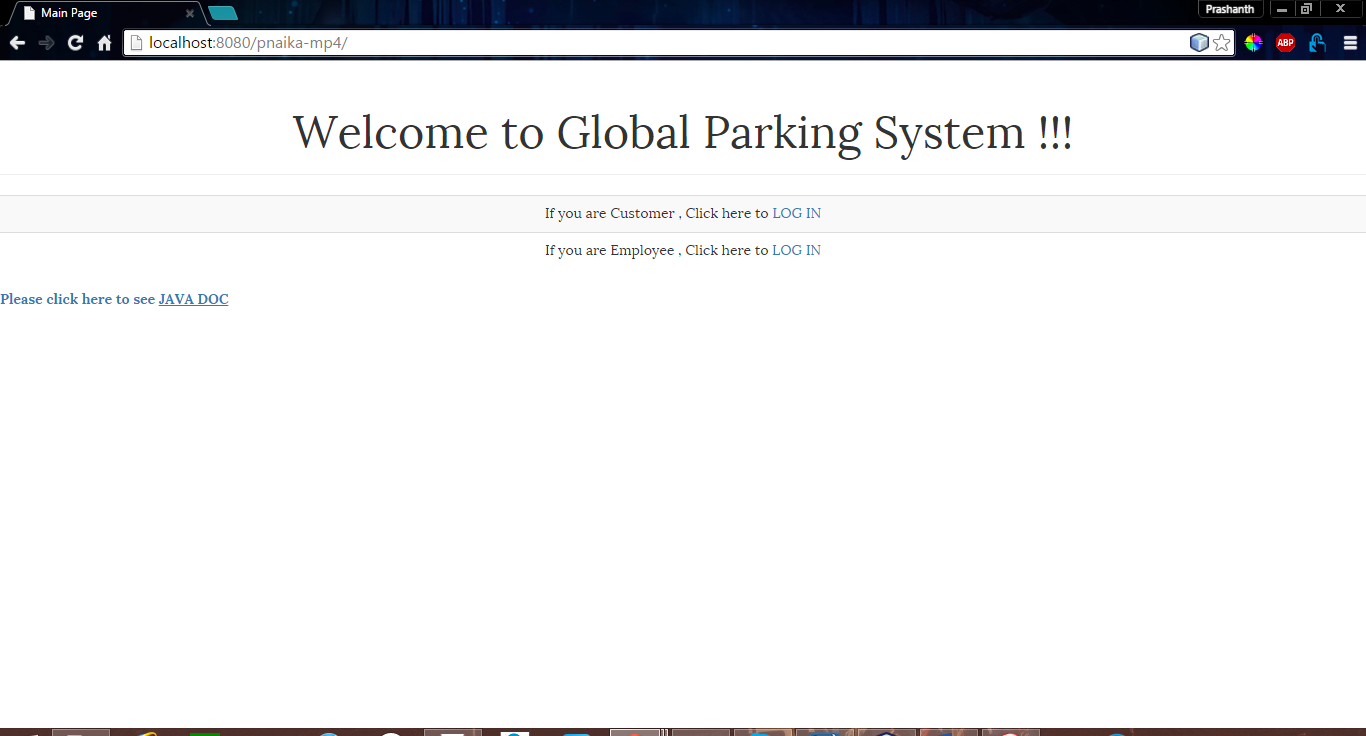


Group-User table



## Screen Shots:

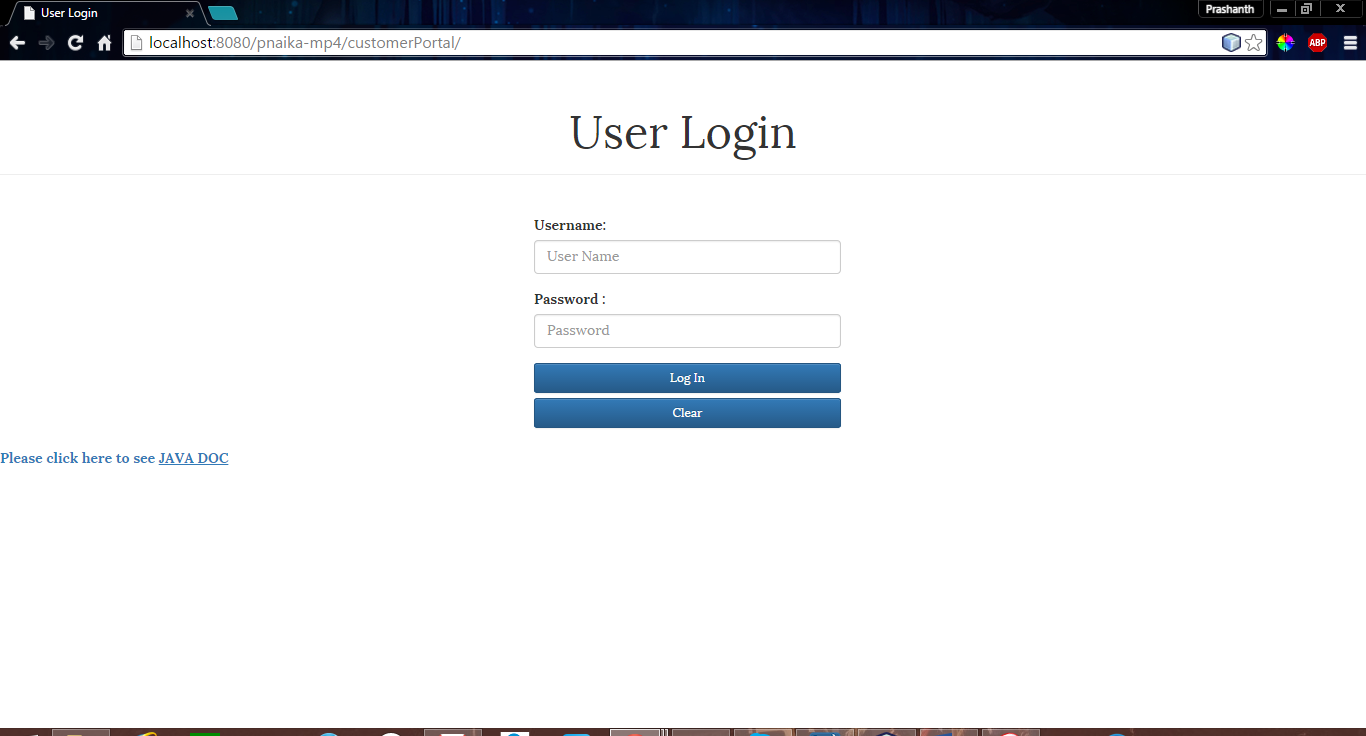
Welcome Page:



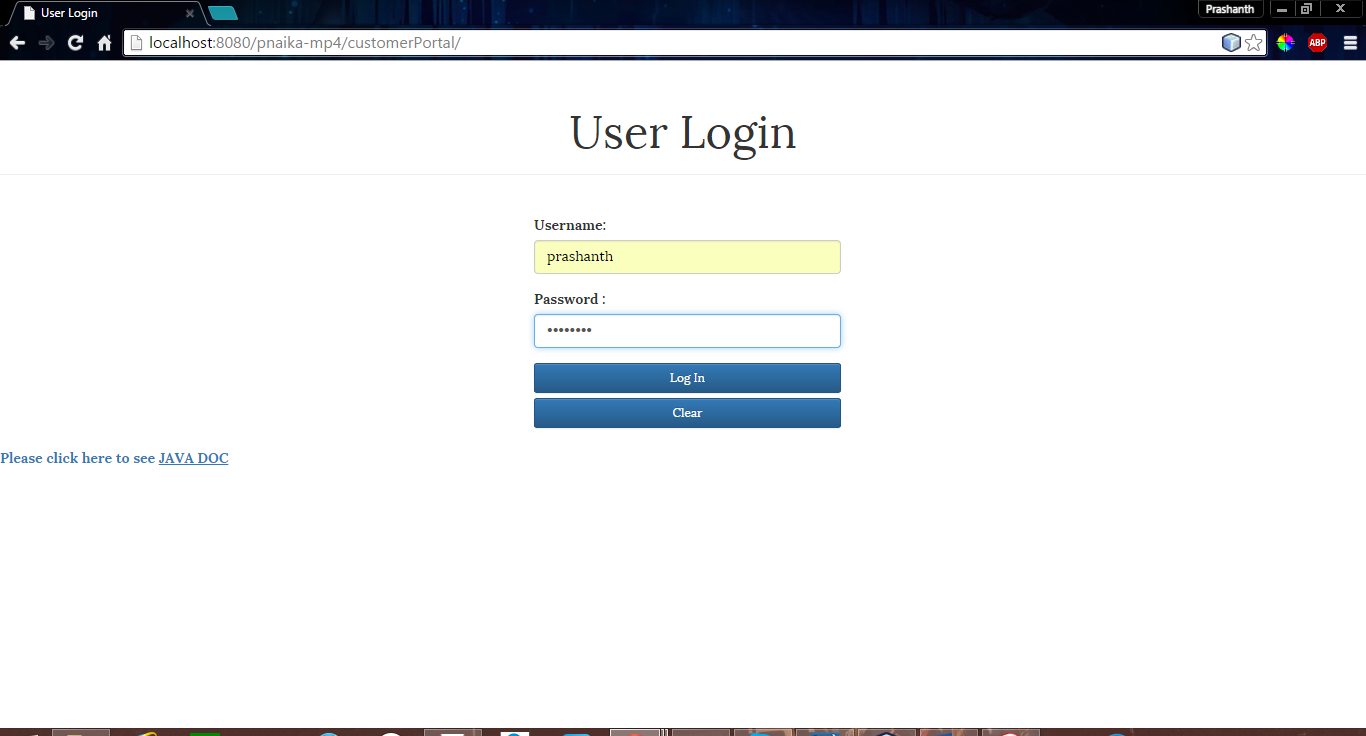
When you Log In as a Customer:

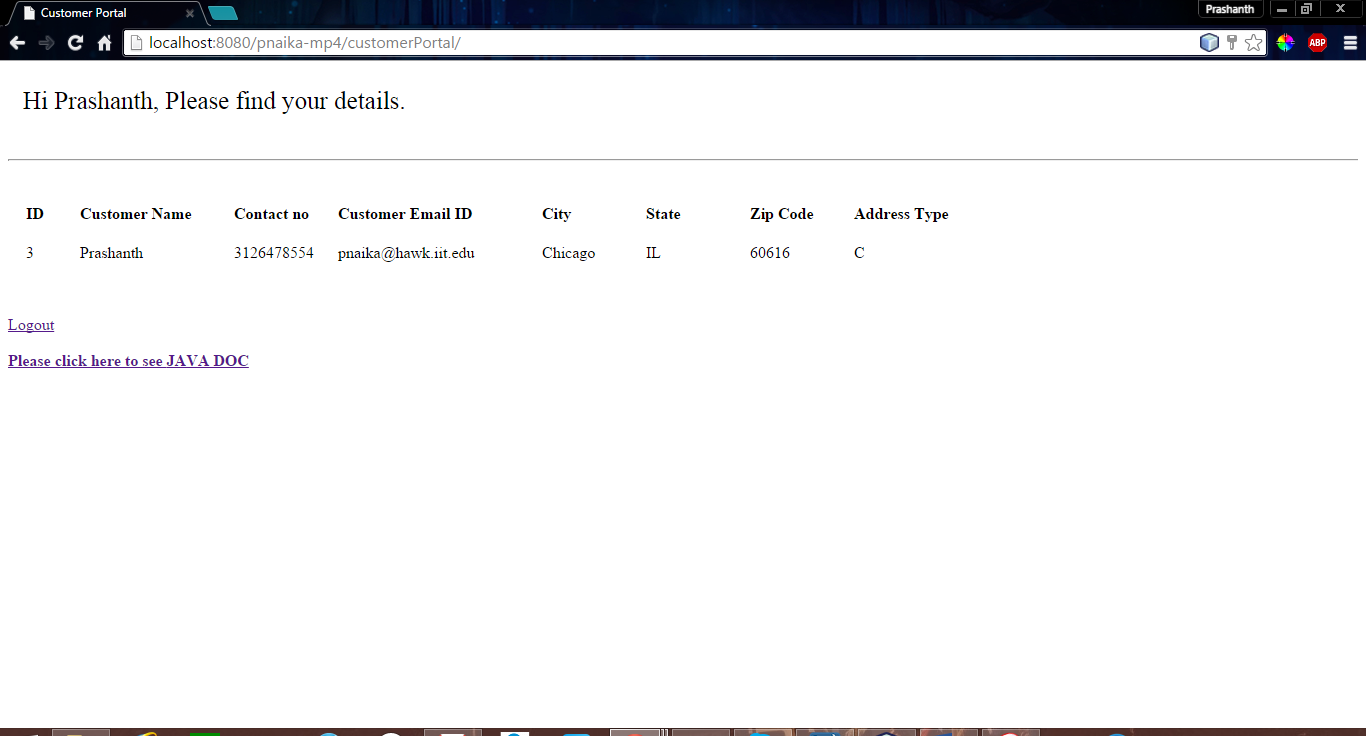
Valid Customer Credentials  **UserName** : prashanth ; **Password** : password

Employee credential ; **USerName :** kesav ; **Password** : password



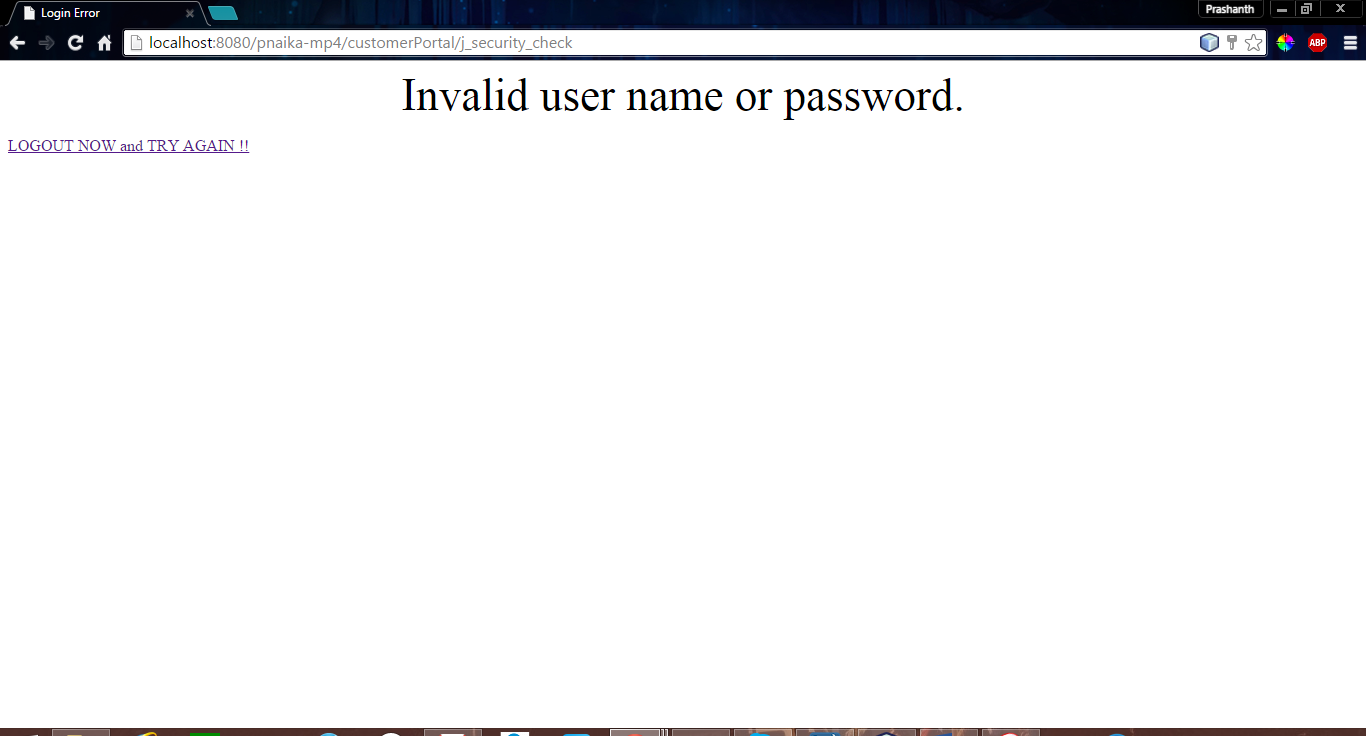
Giving valid input:



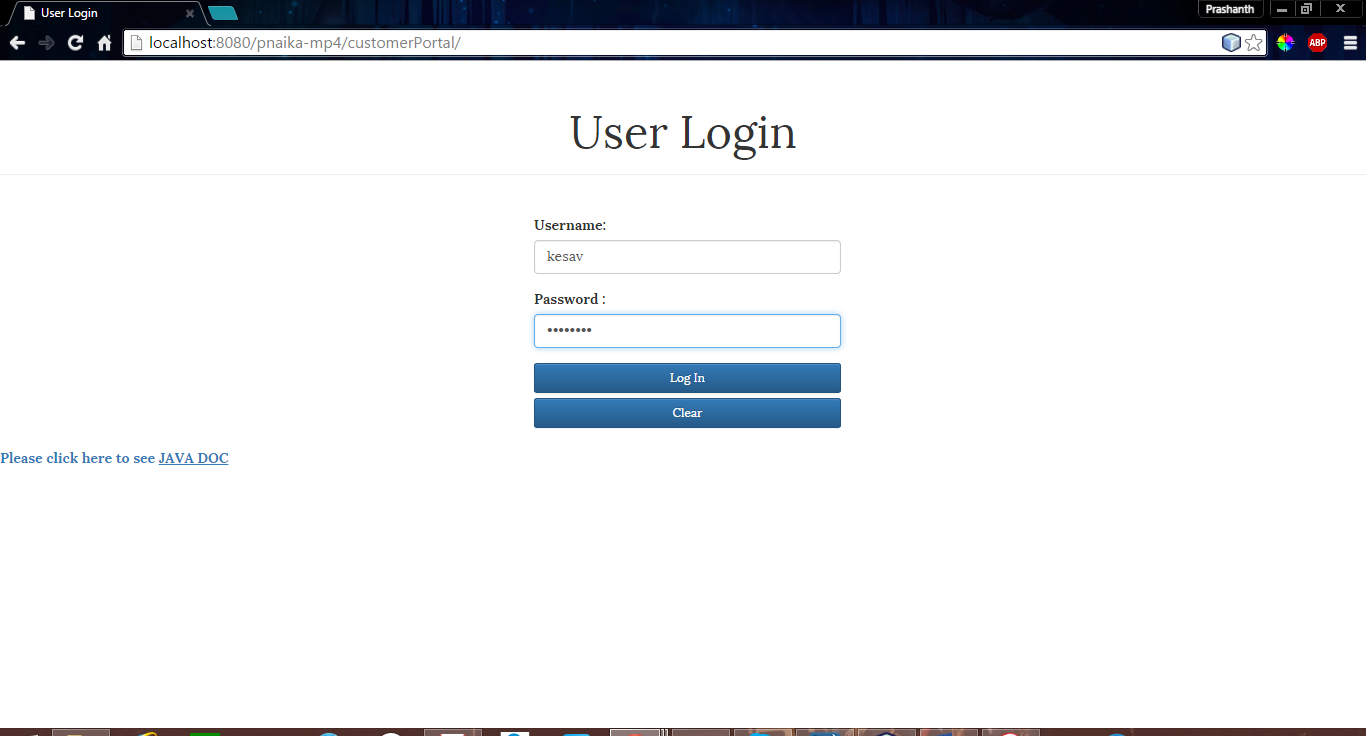


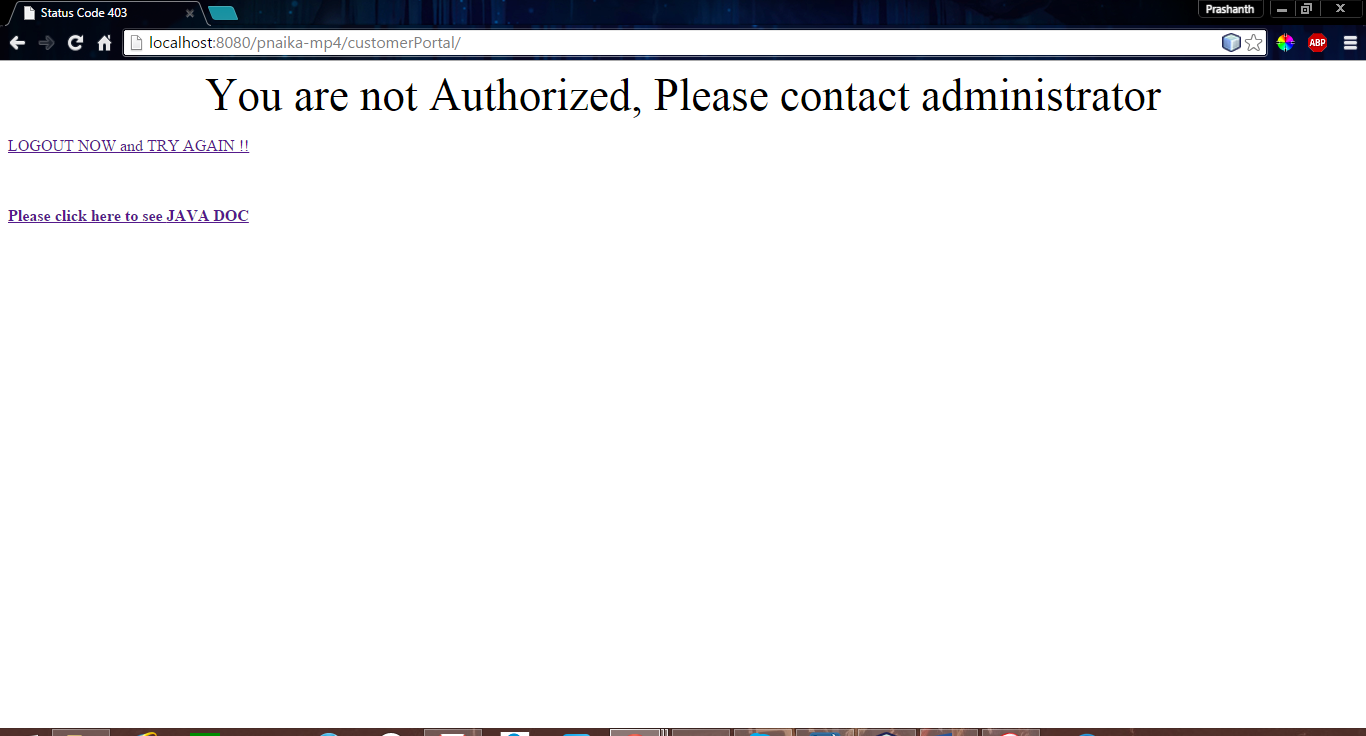
Customer Prashanth can see his complete details.

If Given invalid password then :

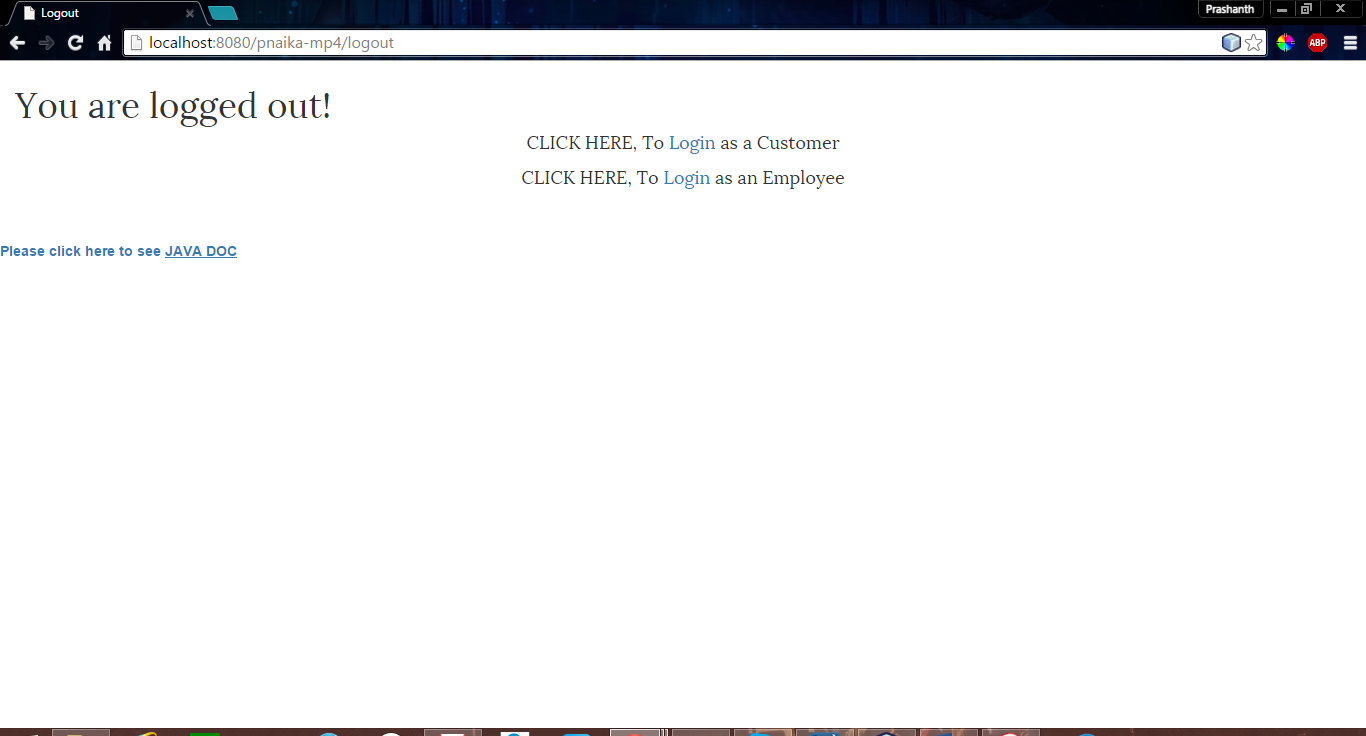


When employee tries to LogIn in Customer Portal:





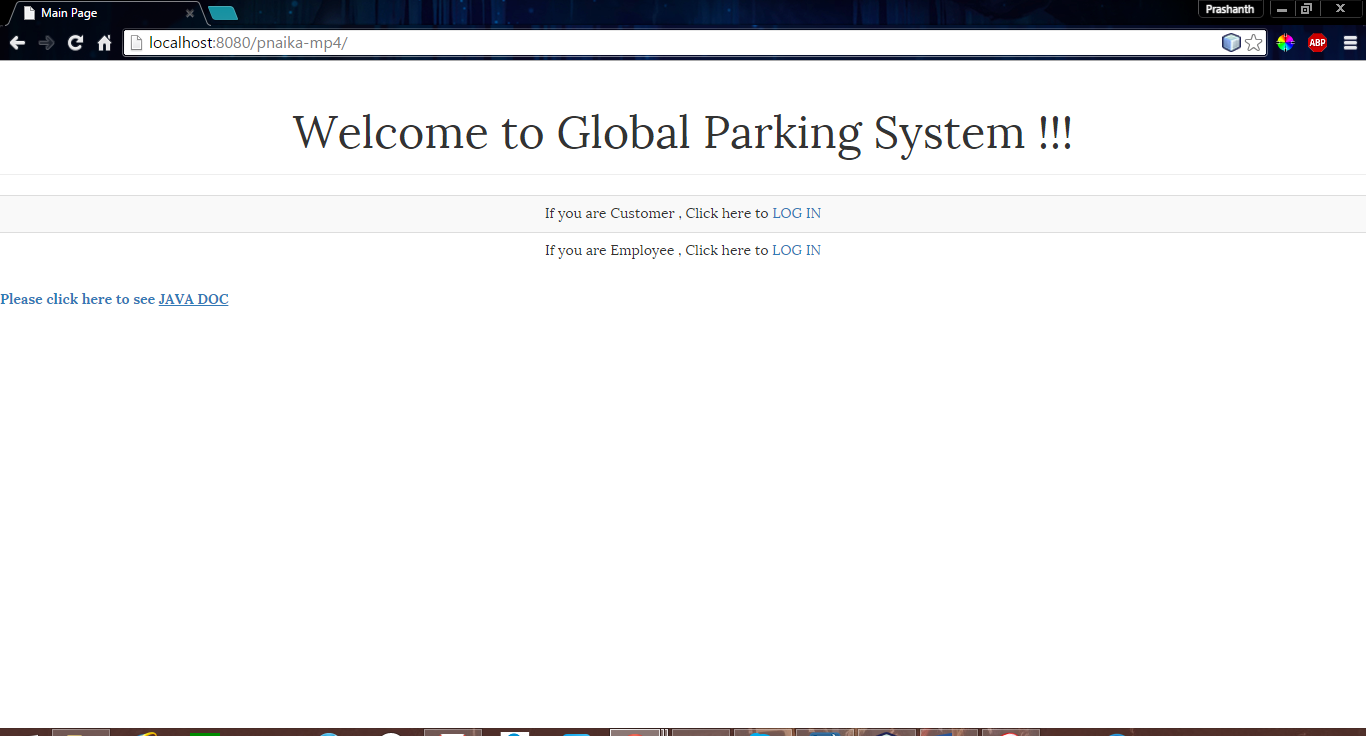
LogOut page is present in all the screen as of now:

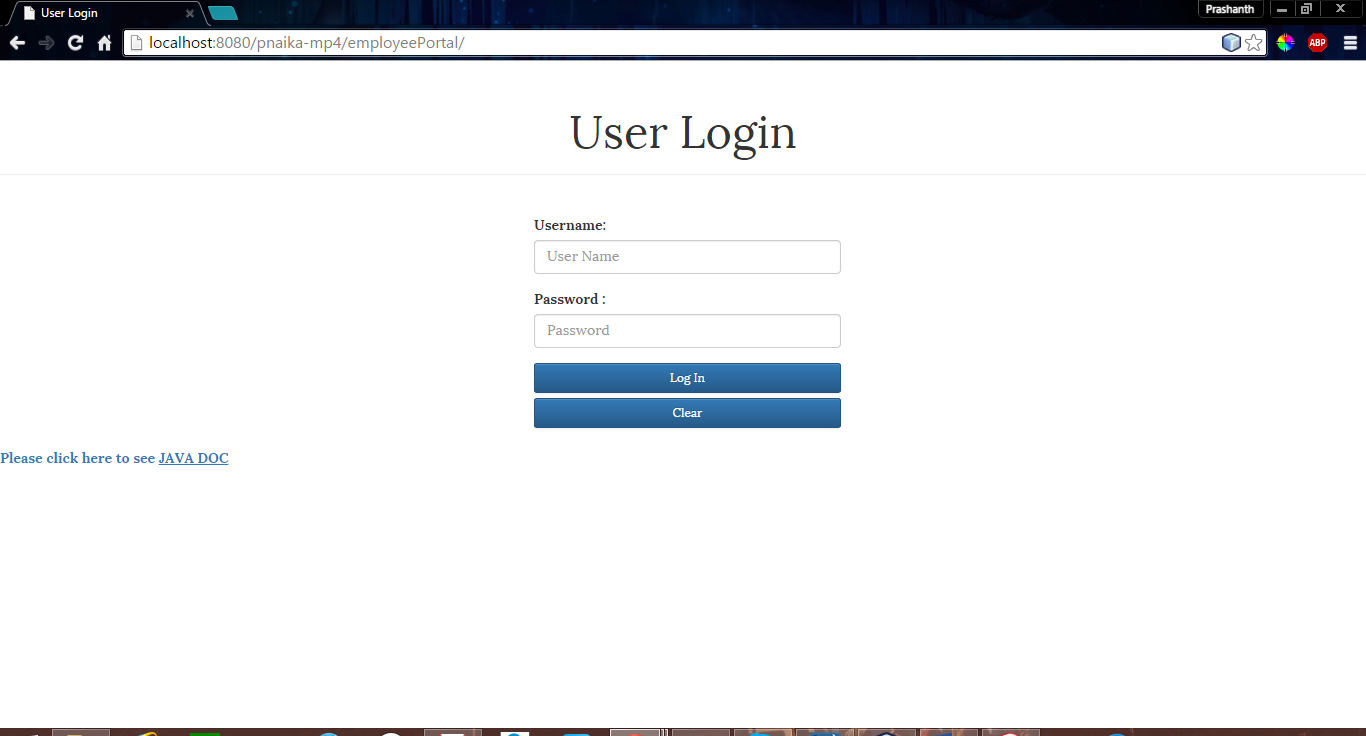


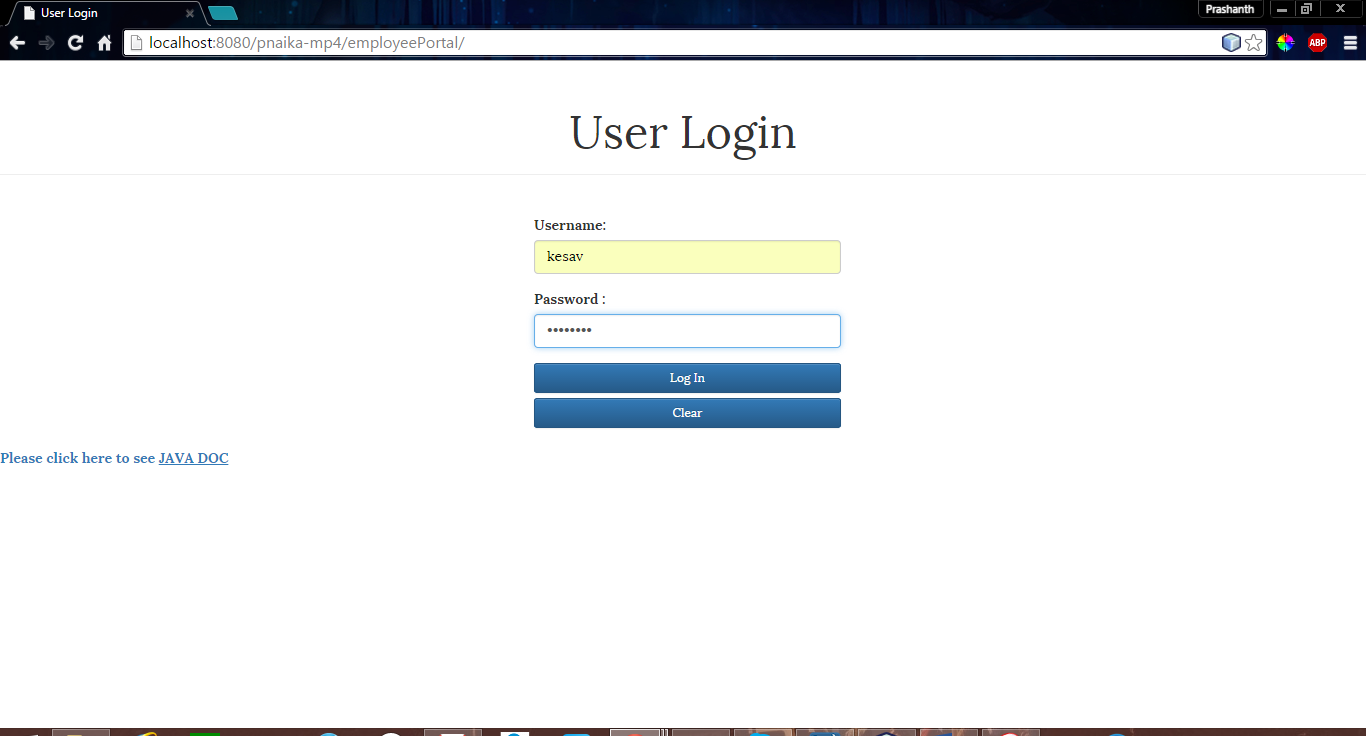
When you Log In as an Employee:

Valid Employee credential ; USerName : kesav ; Password : password

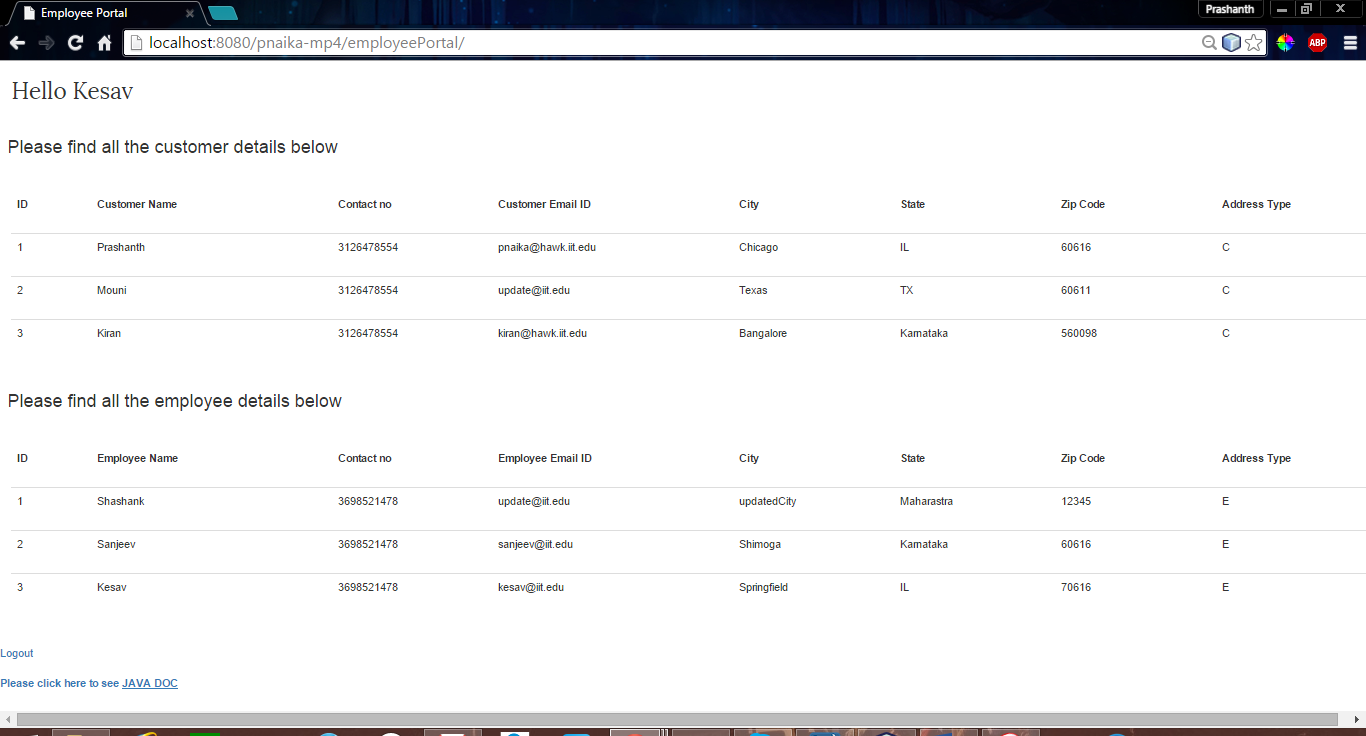
Valid Customer Credentials UserName : prashanth ; Password : password

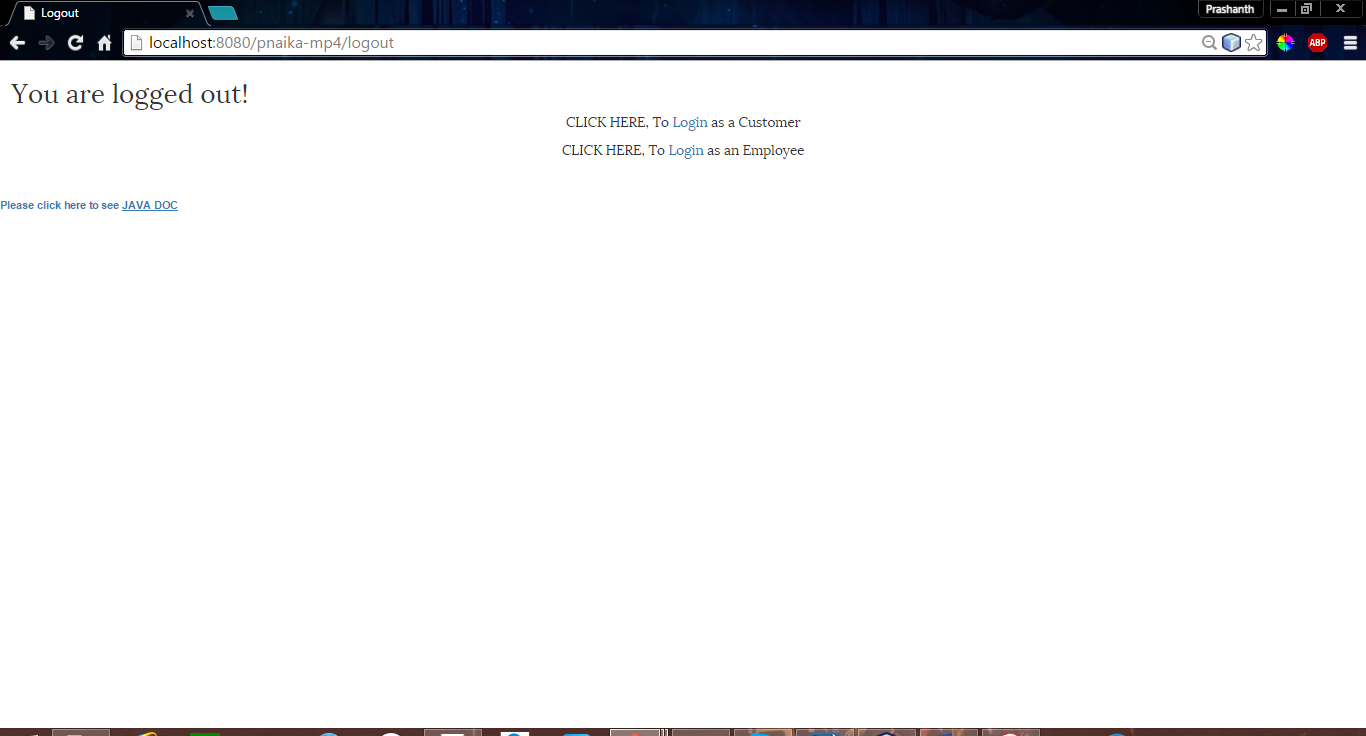




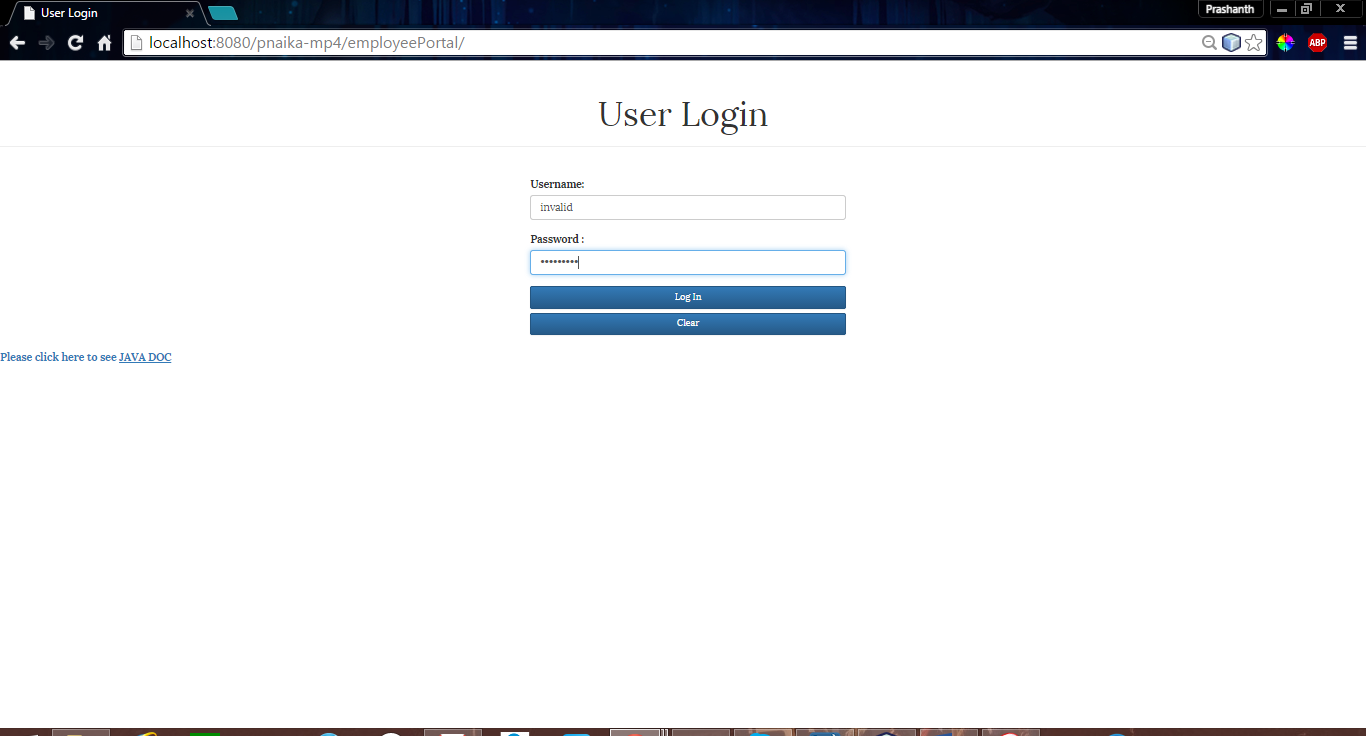


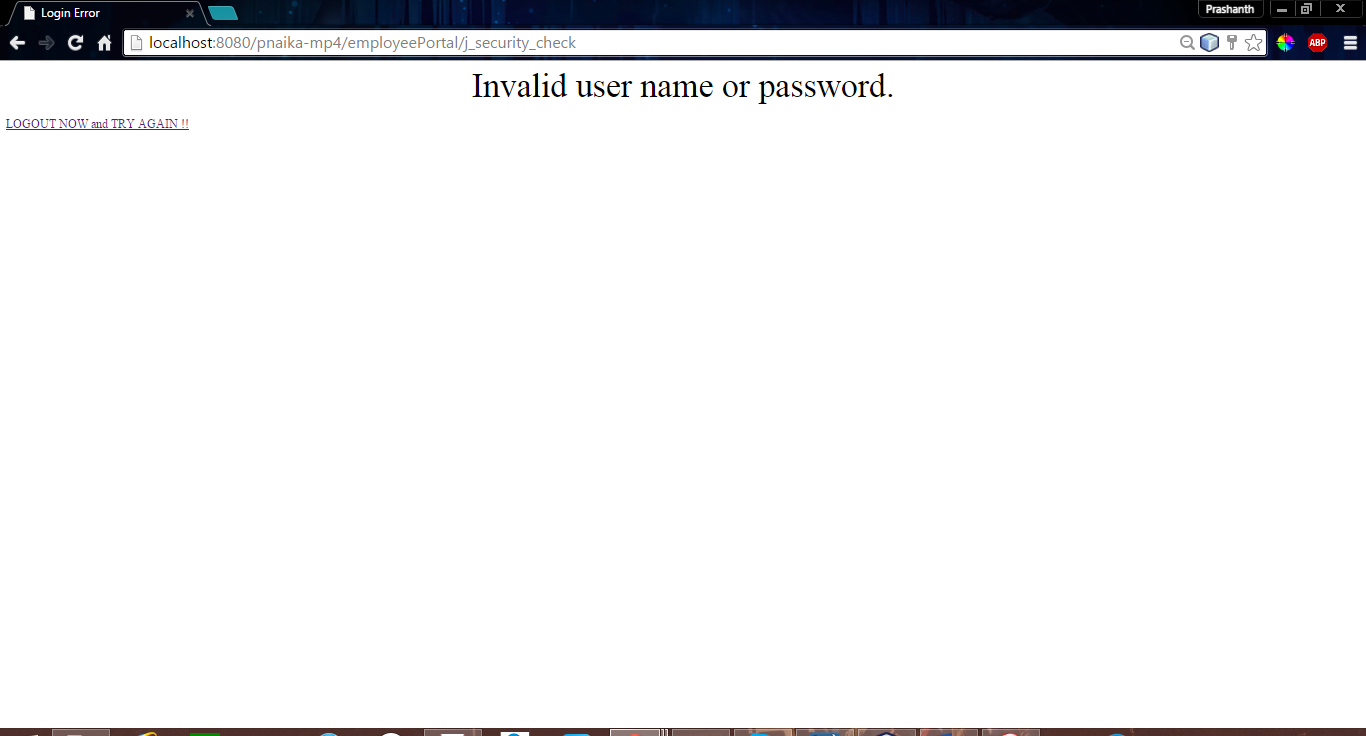
Employee will be able to see all the information of Customers and Employees



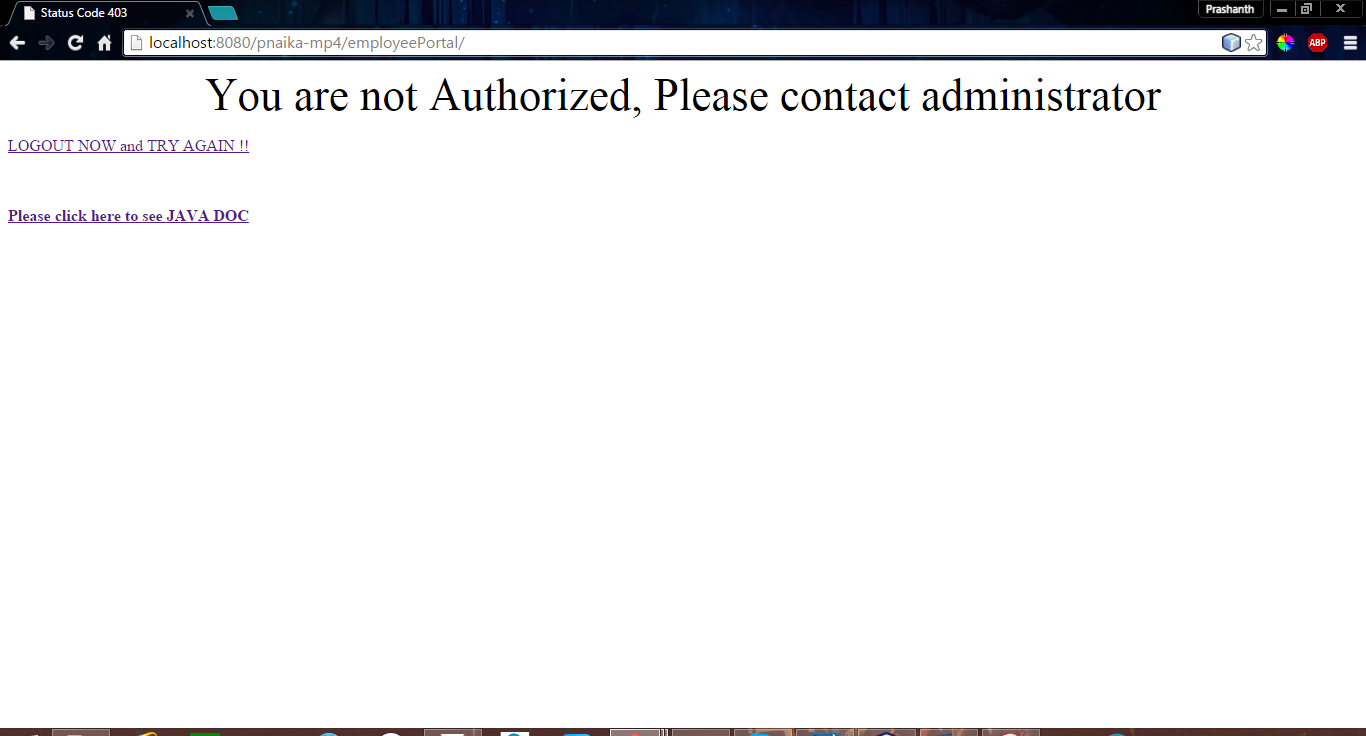


When invalid credential given:

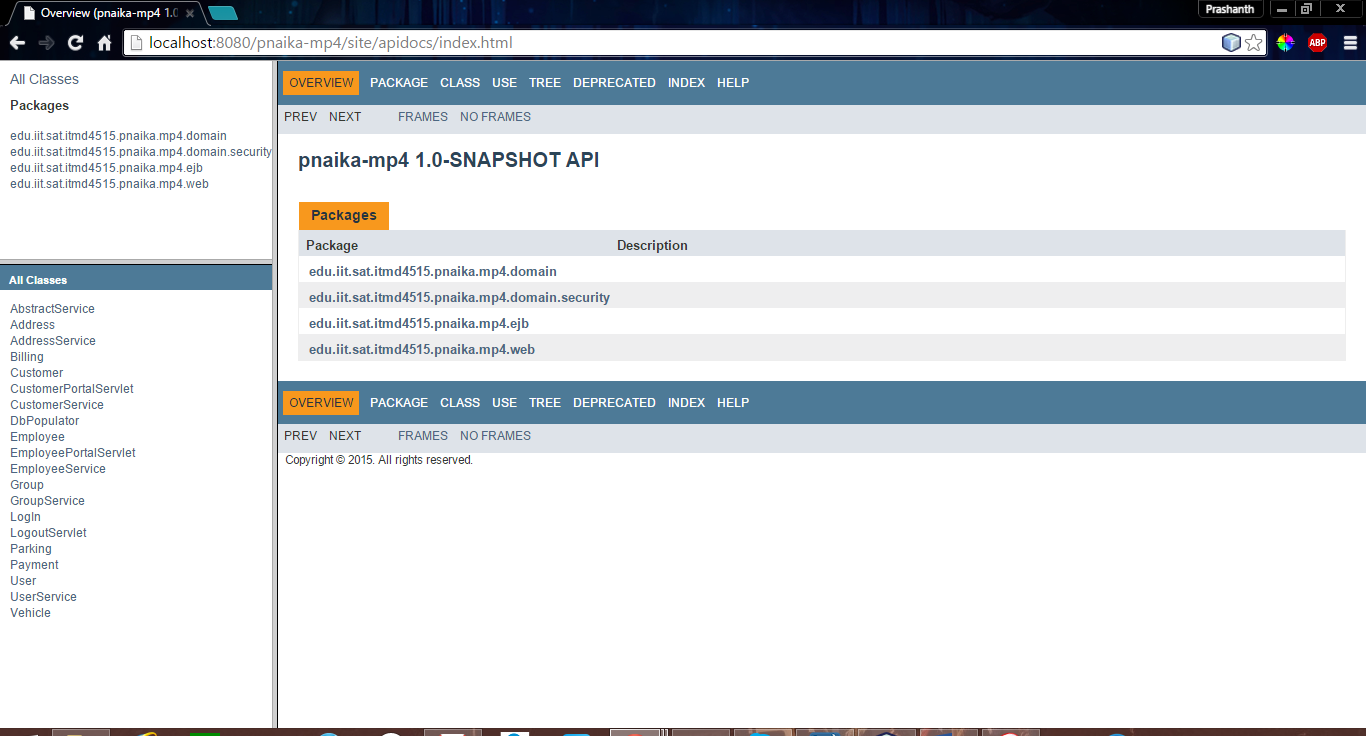




When Customer tries to login in Employee portal:

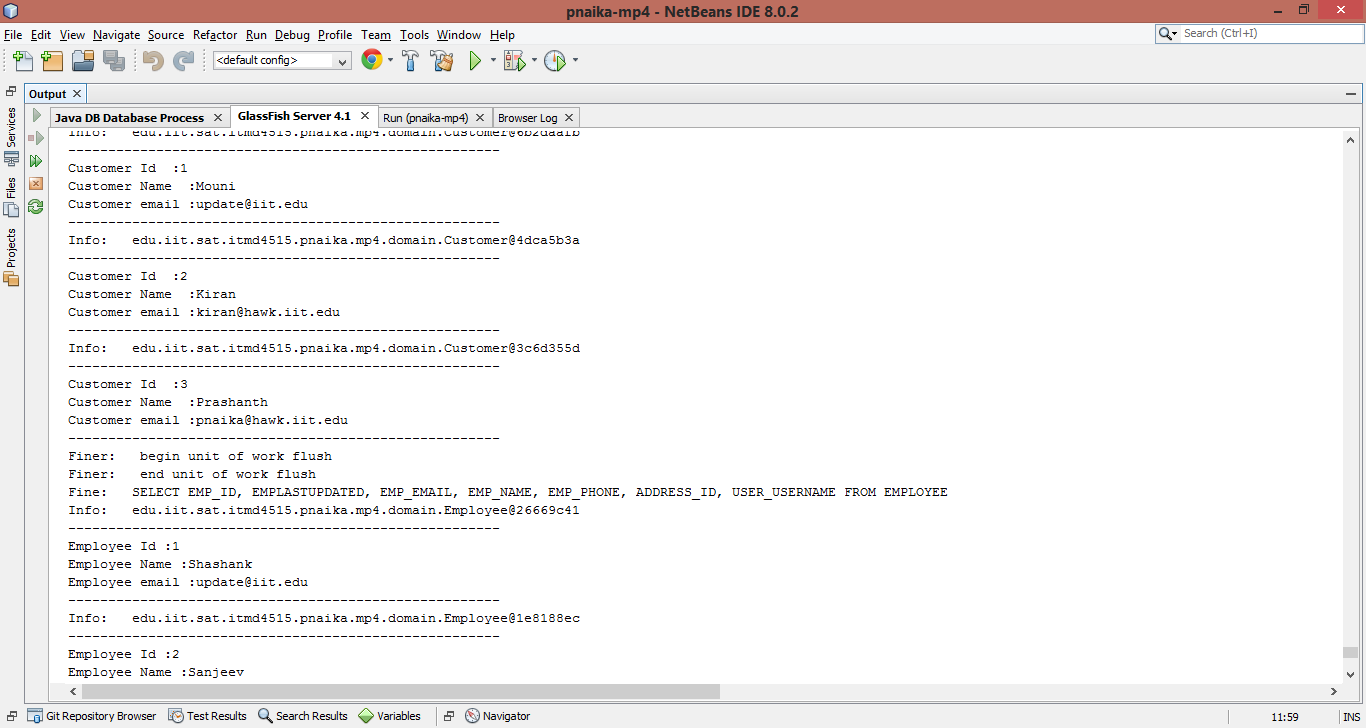


### Java Doc

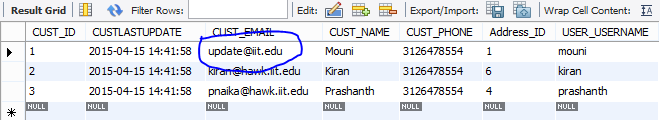


### CRUD method in EJB

Read display in console:



Updated display from MySql



Delete and Create also been implemented in EJB.

## Expected Results:

* Successfully created all the entities and implemented all the relationships.
* Performed data manipulation which insert, update , read and delete test data
* Complete domain model creation and Implementation of security related in web