**CPSC 476**

# Web Back-End Engineering for Enterprise



**Graduate Programming Project**

**GORMANDISE**



#### Submitted by:

## NIYATI BICHU

CWID: 893471748

December 7, 2016

Contents

[Web Back-End Engineering for Enterprise 1](#_Toc468896095)

[NIYATI BICHU 2](#_Toc468896096)

[1. Introduction 4](#_Toc468896097)

[1.1 Purpose 4](#_Toc468896098)

[1.2 Project Description 4](#_Toc468896099)

[1.3 Document Scope 4](#_Toc468896100)

[1.4 Technologies Used 4](#_Toc468896101)

[2. Project’s Architecture 5](#_Toc468896102)

[3. Setup Instructions 5](#_Toc468896103)

[5. Database Schema 6](#_Toc468896104)

[6. Application Working 8](#_Toc468896105)

[7. Issues with the project 13](#_Toc468896106)

[8. Challenges 13](#_Toc468896107)

[9. References 13](#_Toc468896108)

|  |  |
| --- | --- |
|  |  |

## 1. Introduction

### 1.1 Purpose

The purpose of this graduate project is to build a restaurant rating application where users can rate and review restaurants. The users can also post a new restaurant, rate and provide comments and suggestions. Restaurants that are liked have high average rating while those that are disliked are given low rating.

### 1.2 Project Description

The application includes a public page which can be seen as soon as the application starts. A user can only view restaurants and their reviews from the public page. The user cannot add a review or rating from the public page. In order to add a new restaurant or rate an existing one the user needs to login or register. Only a registered user can rate or review a restaurant.

### 1.3 Document Scope

This document describes the functionalities of the Gormandise application. It also describes the configurations and required settings to run it on the system.

### 1.4 Technologies Used

The following technologies have been used for building this project:-

1. **AngularJs for client-side**

AngularJS is a JavaScript framework. It is a library written in JavaScript.

AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag:

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

1. **JAX-RS for server-side-**

 Java API for RESTful Web Services (**JAX**-**RS**) is a Java programming language API spec that provides support in creating web services according to the Representational State Transfer (REST) architectural pattern.

1. **MySQL as database**

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

## 2. Project’s Architecture

## 

In this project, we have used different packages to prevent naming conflicts, to control access, to make searching and locating and usage of classes easier.

1. com.niyati.javaproject.gormandize.model: contains the POJO’s. There are three classes User, Restaurant and Review.
2. **com.niyati.javaproject.gormandize.resources:** contains resource classes- UserResource, RestaurantResource and ReviewResource.

Resource classes are POJOs (Plain Old Java Objects) that are annotated with [@Path](https://jersey.java.net/apidocs-javax.jax-rs/2.0.1/javax/ws/rs/Path.html) have at least one method annotated with [@Path](https://jersey.java.net/apidocs-javax.jax-rs/2.0.1/javax/ws/rs/Path.html) or a resource method designator annotation such as [@GET](https://jersey.java.net/apidocs-javax.jax-rs/2.0.1/javax/ws/rs/GET.html), [@PUT](https://jersey.java.net/apidocs-javax.jax-rs/2.0.1/javax/ws/rs/PUT.html), [@POST](https://jersey.java.net/apidocs-javax.jax-rs/2.0.1/javax/ws/rs/POST.html), [@DELETE](https://jersey.java.net/apidocs-javax.jax-rs/2.0.1/javax/ws/rs/DELETE.html). Resource methods are methods of a resource class annotated with a resource method designator. We use Jersey to annotate Java objects to create RESTful web services.

1. **com.niyati.javaproject.gormandize.service:** contains methods for MySQL database connection and database access and manipulation code.

## 3. Setup Instructions

1. Download and unzip the project on your local drive.
2. The project folder contains a sql script which will create the required schema and the tables.

Run this script in MySQL workbench. If the sql script runs successfully you will find three tables created-

Users, reviews and restaurants.

**The user=root and password=niyati**

The script also inserts two values in the restaurants table so that the public page has some restaurant to display.

1. In the command prompt of the project folder type: mvn clean . This command will clear all the class files. After doing check the target file you should not have anything in it.
2. Run the mvn package this will create a war in the target subdirectory of the project.
3. **Rename the war file to => gormandize.war (mvn package sometimes changes the filename if it does please rename it)**
4. Start the tomcat server by Startup.bat command.
5. Deploy the war file to the tomcat manager or copy the war file in the webapps folder of tomcat.

Go to this url => <http://localhost:8080/gormandize/>

4. Functions:

**UserService**

**addUser()** when a user registers(signs up), the user details are inserted into the MySQL database.

**getUser()** when the user logs in this method is called to check if the username and password entered already exist in the MySQL database.

**RestaurantService**

**addRestaurant()** when a user logs in he can add a new restaurant and also review it. This method inserts a new restaurant(in restaurants table) and also a review(in reviews table).

**getAllRestaurants()** this method fetches all restaurants in the database so that they can be displayed on the homepage.

**ReviewService**

**addReview()** this is called when a user wants to add a review for an existing restaurant on the page.

**getReviews()** this is called when a user wants to view all the reviews for a particular restaurant.

## 5. Database Schema

**Users Table**

CREATE TABLE users (

user\_id INT NOT NULL AUTO\_INCREMENT,

firstname VARCHAR(50) NOT NULL,

lastname VARCHAR(50) NOT NULL,

username VARCHAR(50) NOT NULL,

email VARCHAR(80) NOT NULL,

password CHAR(41) NOT NULL,

PRIMARY KEY (user\_id)

);

**Restaurants Table**

CREATE TABLE restaurants (

restaurant\_id INT NOT NULL AUTO\_INCREMENT,

restaurant\_name VARCHAR(50) NOT NULL,

average\_rating INT NOT NULL,

count INT NOT NULL,

PRIMARY KEY (restaurant\_id)

);

**Reviews Table**

CREATE TABLE reviews (

review\_id INT NOT NULL AUTO\_INCREMENT,

user\_id INT NOT NULL,

restaurant\_id INT NOT NULL,

review VARCHAR(100) NOT NULL,

rating INT NOT NULL,

PRIMARY KEY (review\_id),

FOREIGN KEY (user\_id) REFERENCES users (user\_id),

FOREIGN KEY (restaurant\_id) REFERENCES restaurants (restaurant\_id)

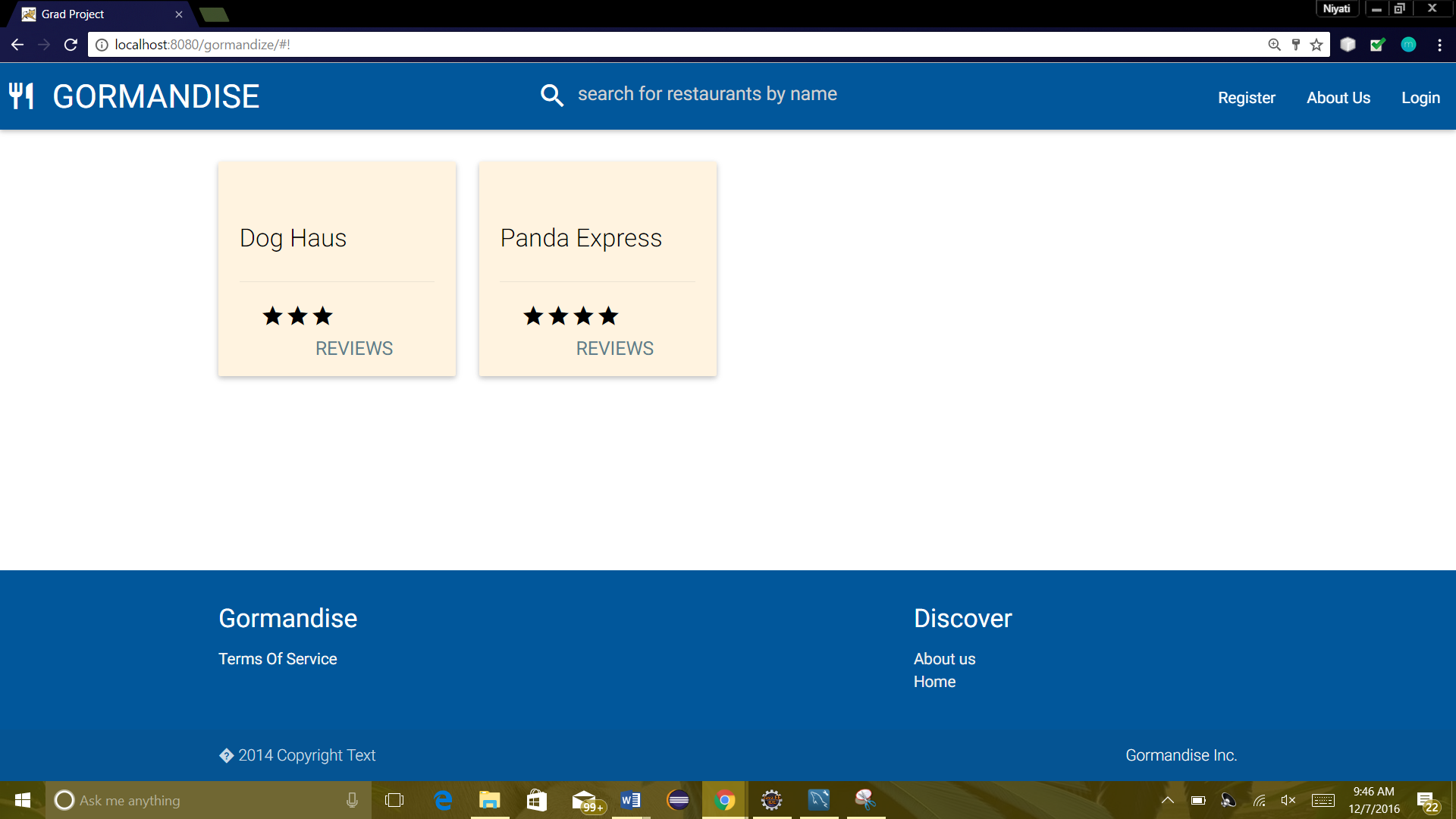
);

**Pom.xml:** contains all the dependencies required for

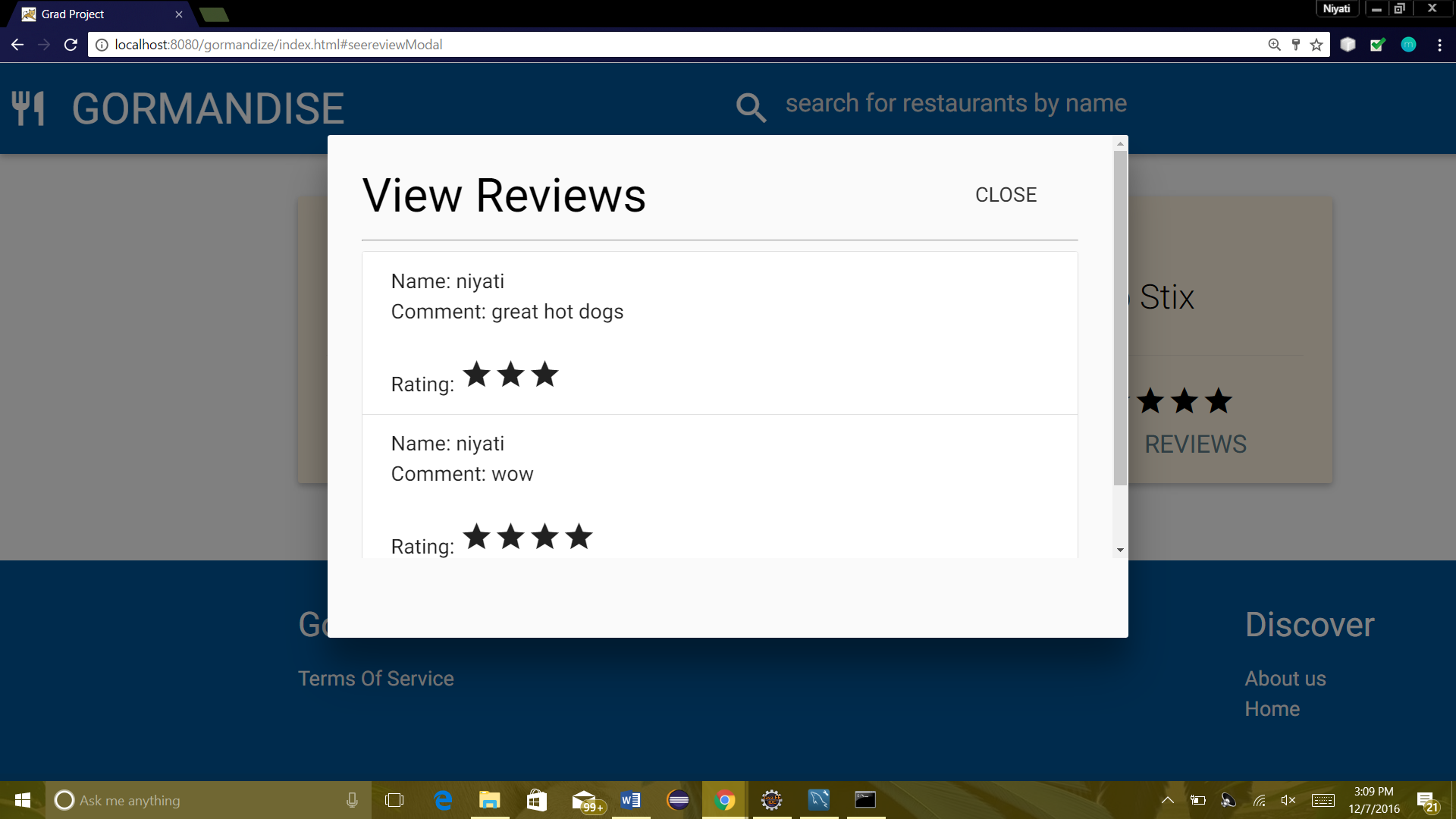
1. Jersey
2. MySQL

## 6. Application Working

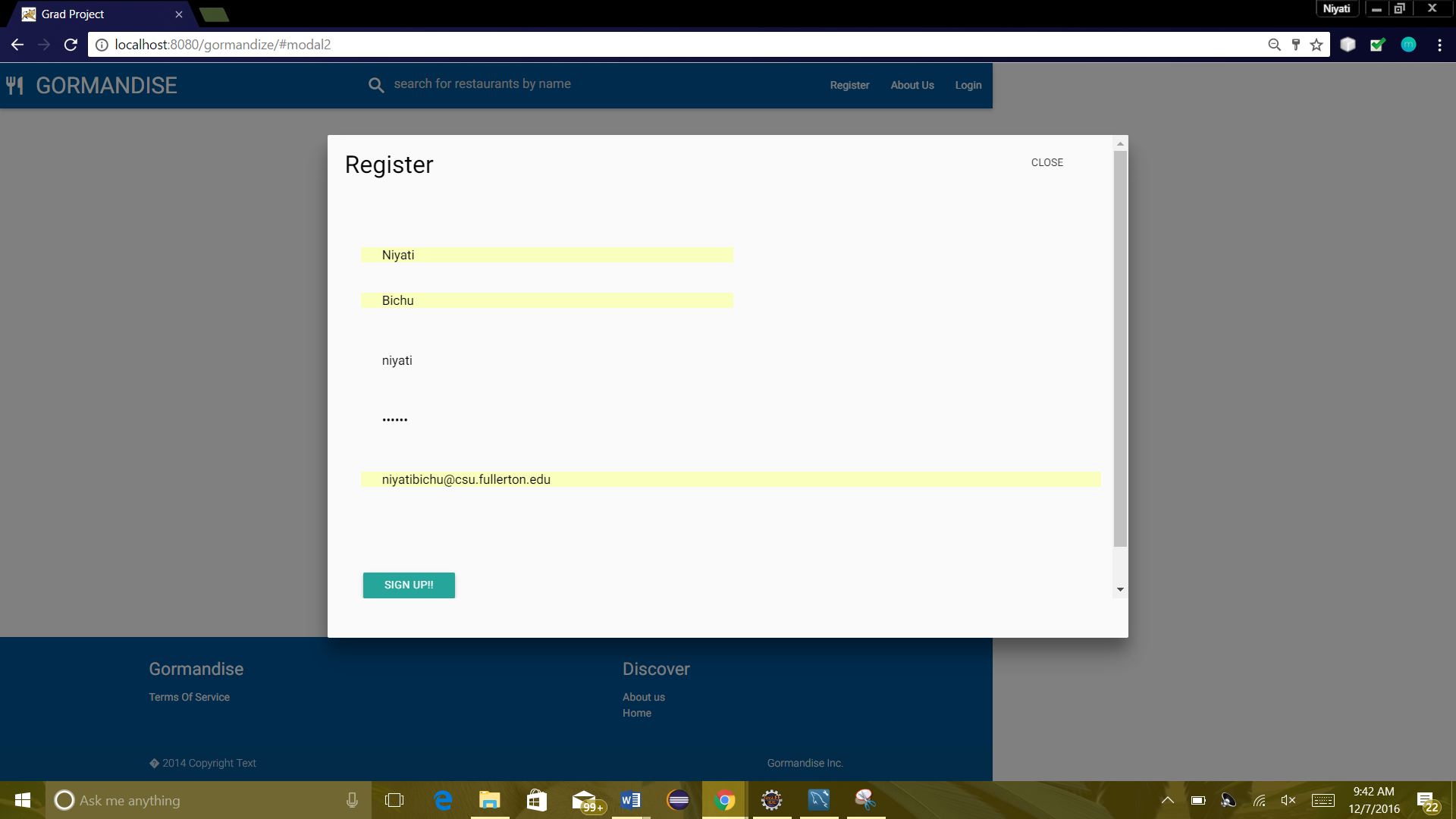
The homepage/public page of Gormandise is shown below. This is what we see when the application starts.



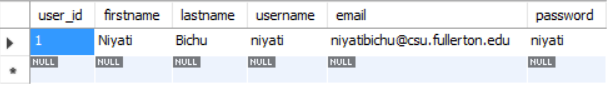
On the public page a user can only view restaurants and their reviews. When we click on **REVIEWS** in any of the cards we can see the reviews as shown below-



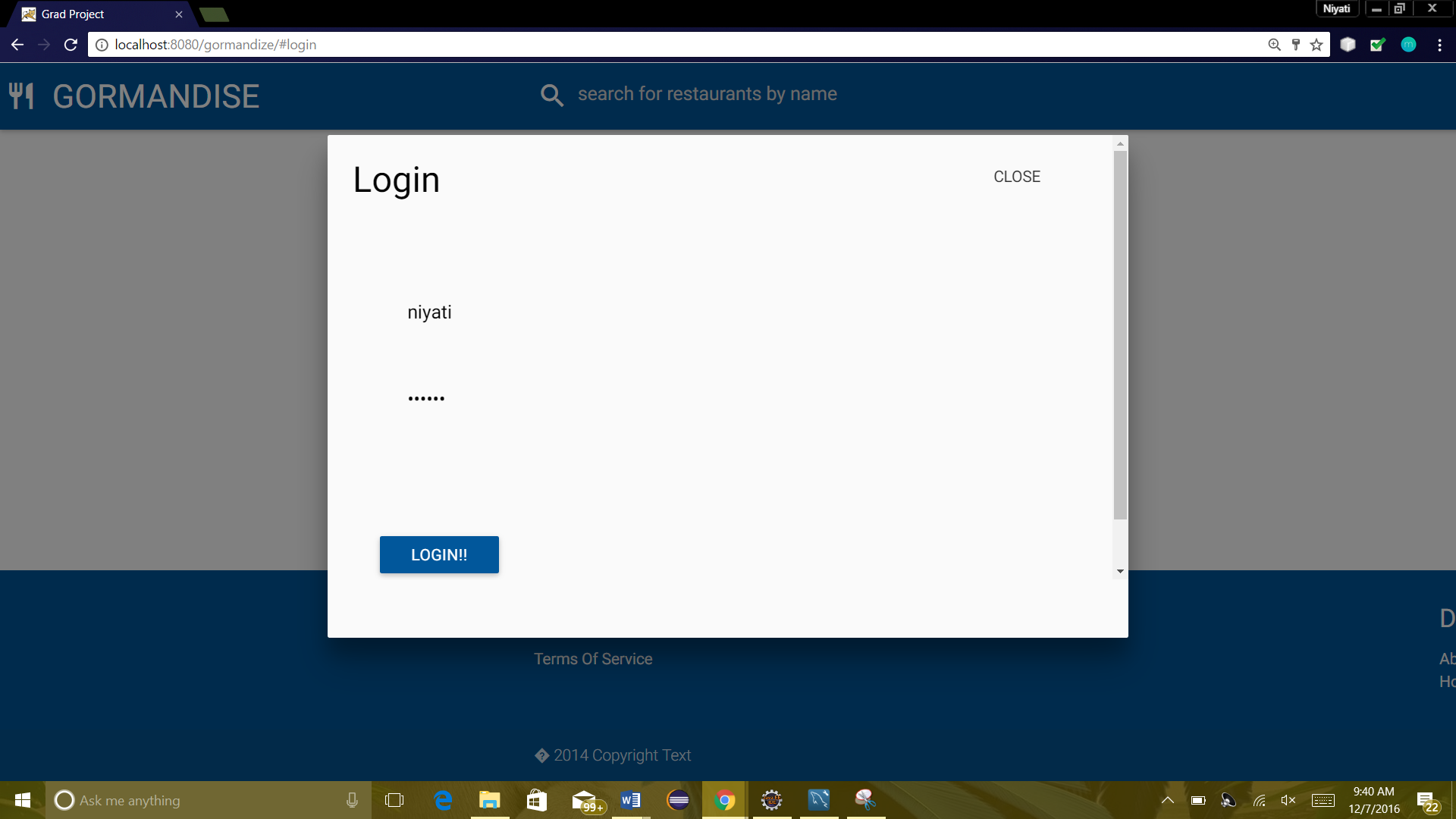
Users can sign up by providing details such as first name, last name, username, email and password. Validation is done for each of these fields.



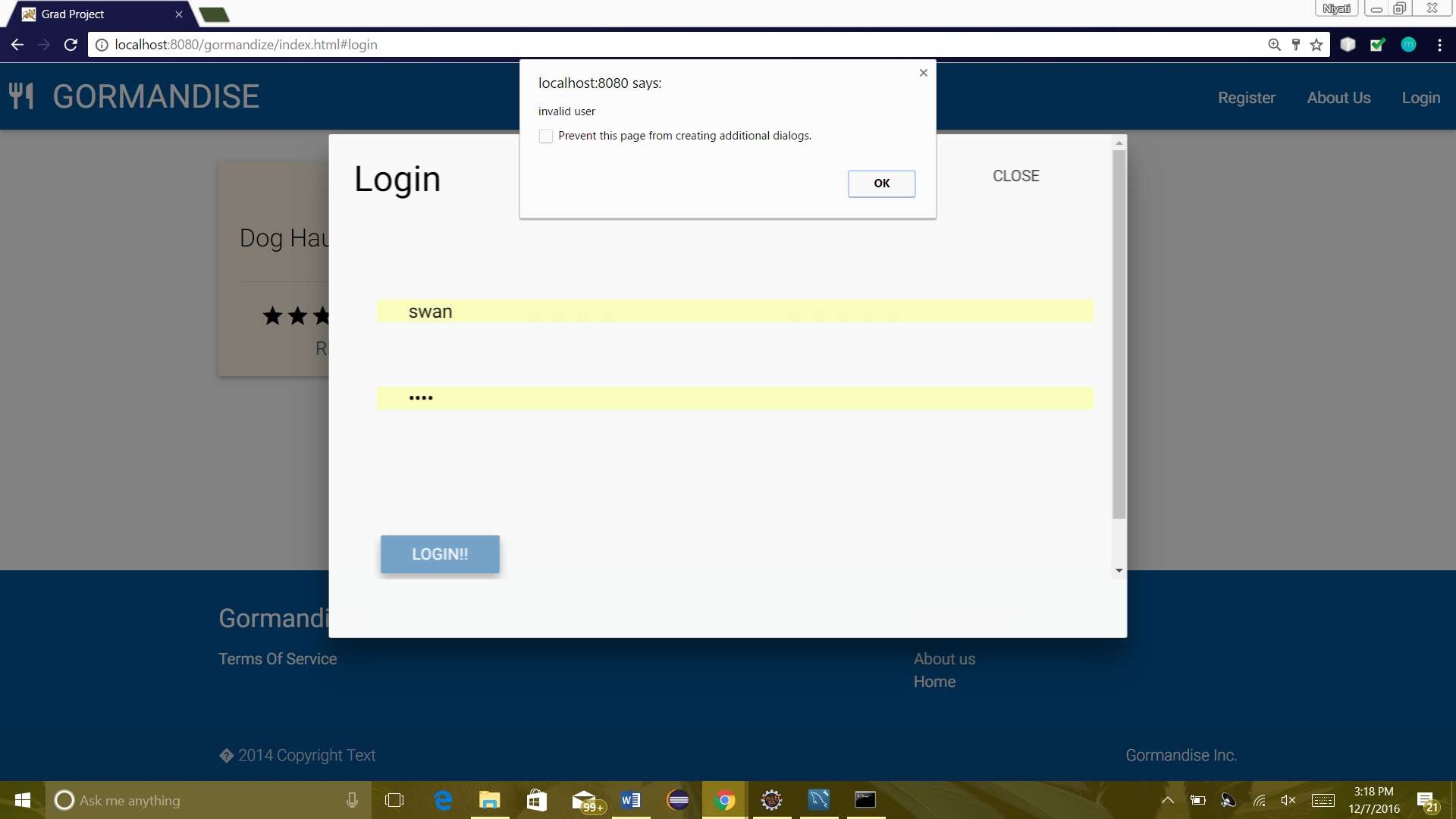
If the user registers successfully then the user object is inserted in MySQL database of users as shown below.



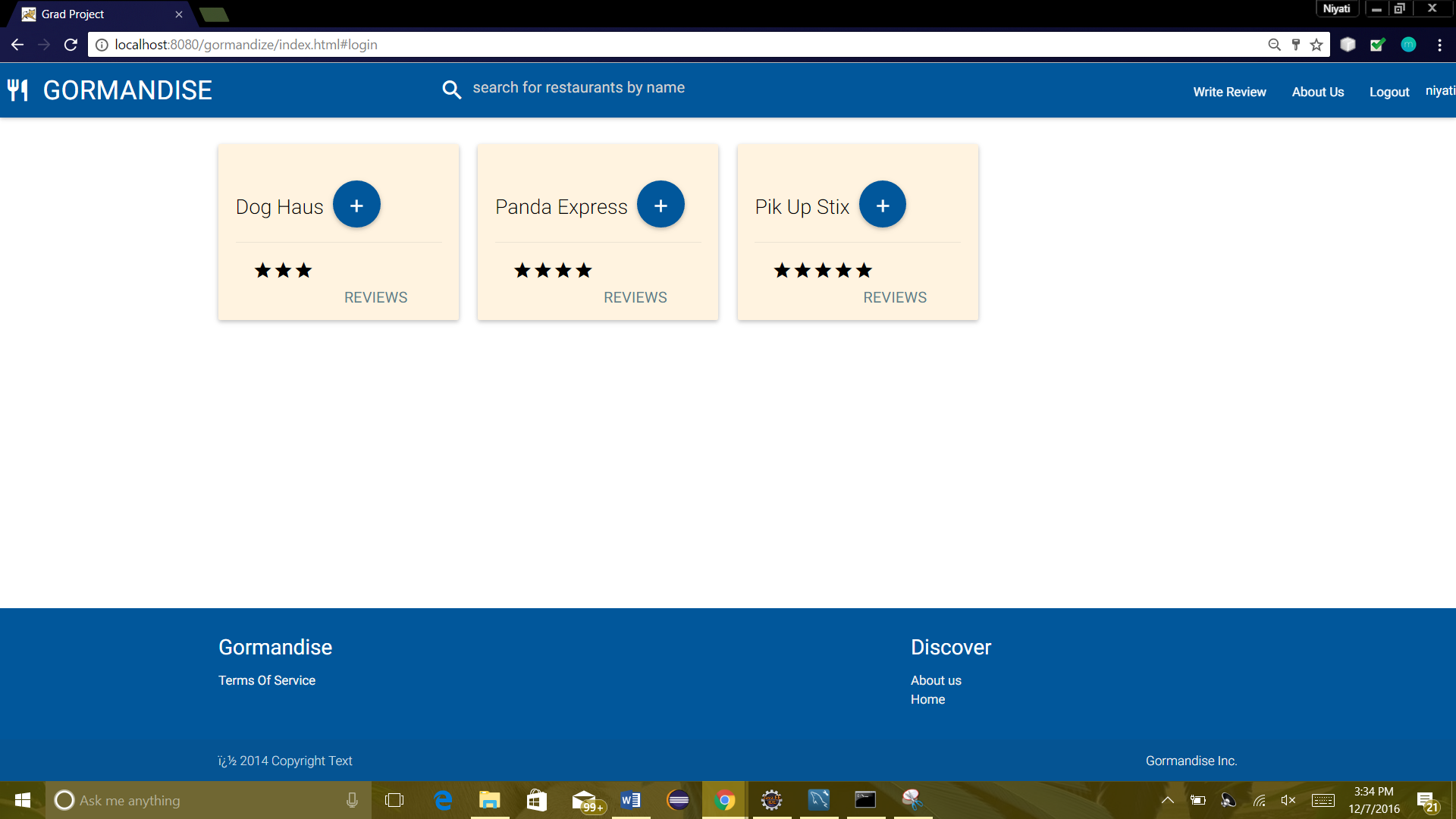
Users can login by entering their username and password.



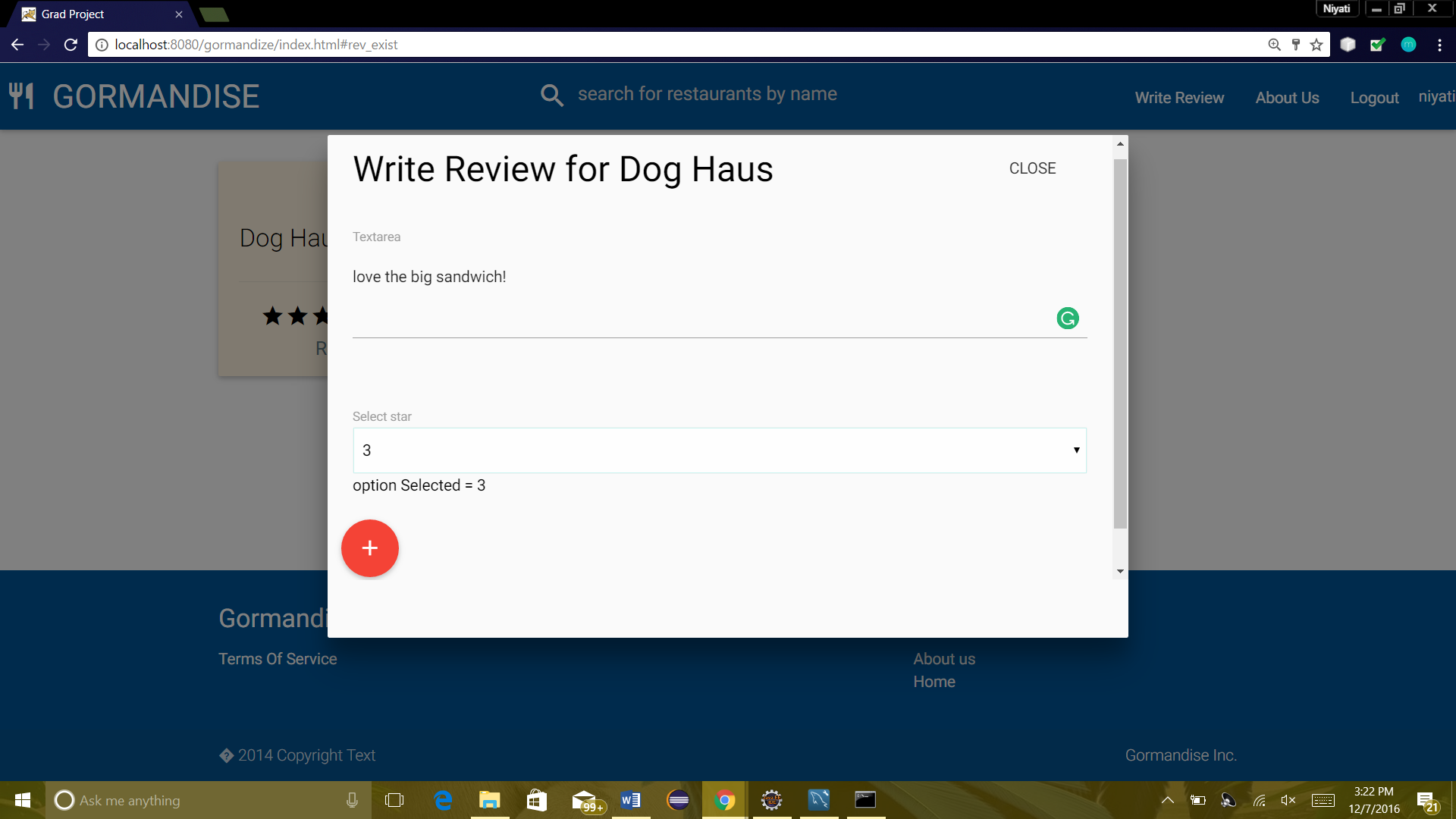
If the login details entered are incorrect then an alert is popped up saying INVALID USER



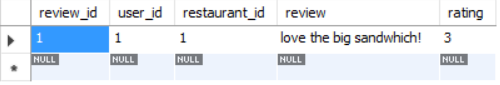
If the login details enter are correct the user is logged in and can see a page as shown below. On this page user can rate and review restaurants.



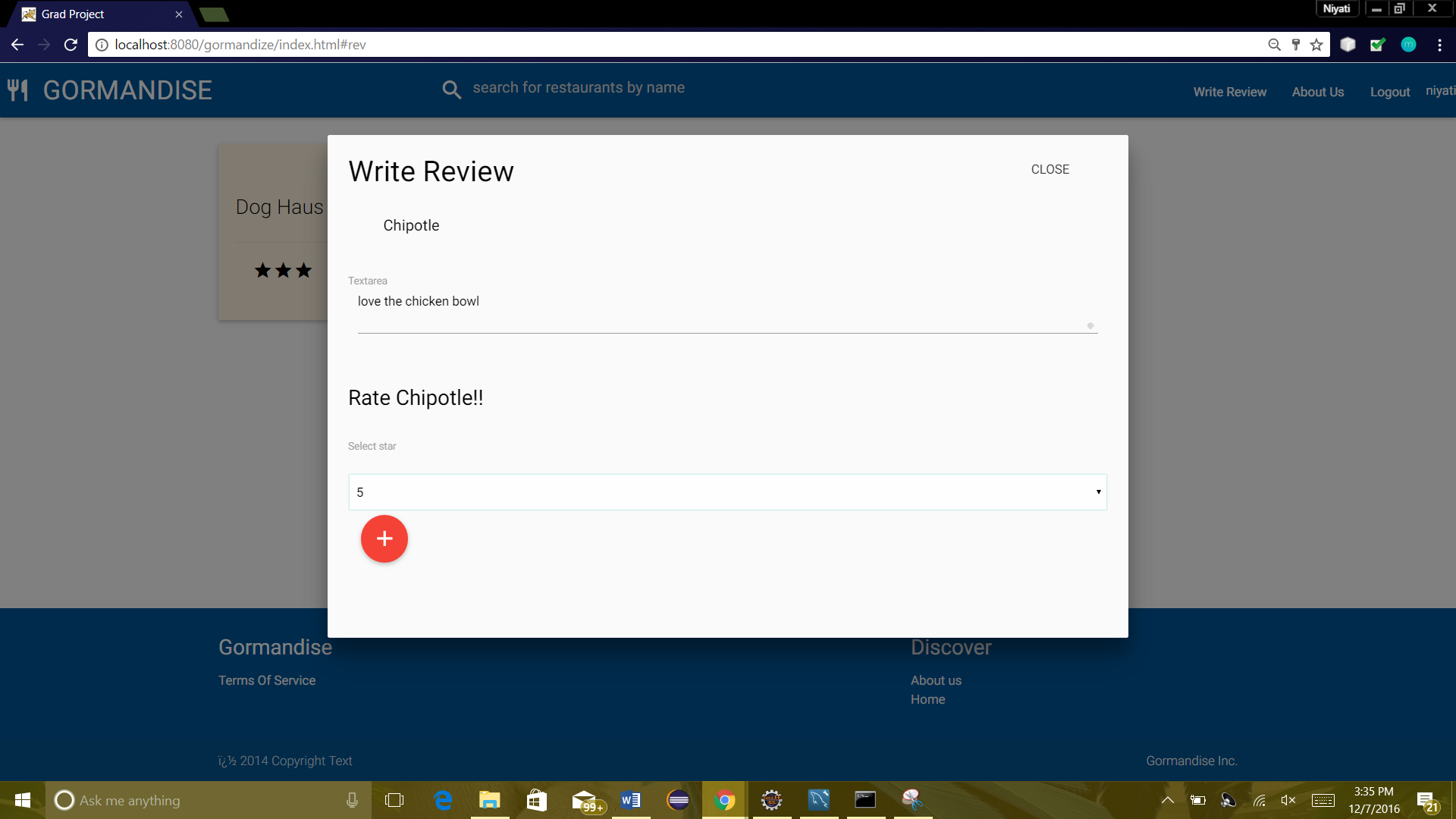
A user can review and rate an already existing restaurant by clicking the add button of the restaurant he wishes to rate. On clicking the add button user will see the below page-



When the red add button is clicked this review will be added to the review table



The user can also add a new restaurant and rate it



When the user clicks on the red add button the restaurant is added to the database as shown below-



The user can logout after he is done using the application

## 7. Issues with the project

1) When a new restaurant is added, after clicking the add button the user cannot see the restaurant on the page unless the page is refreshed.

## 8. Challenges

The challenges faced during the project were:

1. Problems faced in MySql query syntax.
2. Sending data from AngularJS to JAX-RS was tedious. But JSON format made it easy

## 9. References

1. REST with Java(JAX-RS) using Jersey-Tutorial

<http://www.vogella.com/tutorials/REST/article.html#jerseyprojectsetup>

1. Developing RESTful APIs with JAX-RS

<https://www.youtube.com/watch?v=xkKcdK1u95s&list=PLqq-6Pq4lTTZh5U8RbdXq0WaYvZBz2rbn>

1. Java API for RESTful Web Services

<https://en.wikipedia.org/wiki/Java_API_for_RESTful_Web_Services>

1. MySQL Tutorial

<https://www.tutorialspoint.com/mysql/index.htm>