**Project Description**

**【 GENERAL INFORMATION 】**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| App Title | My Dream Instructor | |  |  |
|  |  | |  |  |
| Group Name | DreamTeam | |  |  |
|  |  | |  |  |
| Group Member | Dikshit Sharma | | 300287961 |  |
|  | Robin Singh Sidhu | | 300292039 |  |
|  | Viacheslav Sierkov | | 300287665 |  |
|  | |
|  |  | |  |  |
| Course Title | CSIS 3280 Web Scripting | | |  |
|  |  | |  |  |
| Submitted Date | Apr 14, 2020 | |  |  |
|  |  | |  |  |

**【 OVERVIEW 】**

My Dream Instructor behaves like a mini-ratemyproffesor website which allows students to register, login, search for an instructor, add their experience about the instructor and a particular course that instructor teaches, and leave their rating from 1 to 5. Students are also able to edit their reviews and delete them accordingly. For the statistics, our website calculates average rating for the particular instructor, which is calculated based on student’s reviews, and shows total number of reviews. From the admin perspective, all the instructors, courses, courses that particular instructor teaches, and students are listed. An admin is able to add a new instructor, course, course that instructor teaches, edit, and delete respectively. From the side of the security, sessions are handled, and passwords are encrypted. Input validation is implemented for searching instructor, registration, login, and creating review. Web Service using JSON is integrated within Professors page, Laravel for Logout Page is implemented, and CodeIgniter for About Us page is implemented accordingly.

**【 FEATURES WITH ADDITIONAL TECHNOLOGIES 】**

|  |  |  |
| --- | --- | --- |
| **No** | **Features** | **Description** |
| 1 | Login/Registration | A user will be able to register and login to an account and check a button that will save his login and encrypted password information in the database. |
| 2 | Search for Instructor | A user will be able to search for particular instructor according to their full name. In order to help the student to find the instructor, autofill is implemented using JavaScript. |
| 3 | View Instructor Reviews | After instructor is found, a user will be able to see all instructor reviews from all the students that had left it previously, with all the information like student name, date, course name and their experience. A user will be also able to see average rating and number of reviews for particular instructor. |
| 4 | Create, Edit, and Delete Reviews | After instructor is found, a user can create his own review for that instructor. He/she has to specify a rating, select a course for giving rating to, and leave an experience for that course and instructor. After creating a review, he/she can see a newly created rating, which furtherly may be modified or deleted. |
| 5 | Admin Page | For the ease of manipulating the entities, admin Page provides an admin with all the entities he/she may manipulate. CRUD operations for the entities are implemented as well. |
| 6 | Laravel | For the first technology, our team implemented a logout page using Laravel. |
| 7 | CodeIgniter | For the second technology, our team implemented an about us page using CodeIgniter. |
| 8 | UI Bootstrap | For all the UI pages, Bootstrap CSS had been used to solidify the good look of our app. |
| 9 | Web Service using JSON | Web Service is implemented that integrates with Professors page, using RestAPI, and RestClient. |

**【 ENTITIES RESPONSIBILITY 】**

|  |  |  |
| --- | --- | --- |
| **No** | **Features** | **Assignees** |
| 1 | Student Entity | Viacheslav, Dikshit, Robin |
| 2 | Instructor Entity | Viacheslav, Dikshit, Robin |
| 3 | Review Entity | Viacheslav, Dikshit, Robin |
| 4 | Course Entity | Viacheslav, Dikshit, Robin |

As our app works like a real high-component system and all the entities interacts with each other, all the group members had to work with all the entities, and link them between each other to achieve high quality app. Everyone contributed to managing the entities structure and communication between each other using foreign keys and intersection tables.

**CLASS DIAGRAM**

A close up of text on a white background

Description automatically generated

**Meeting Minutes**

Since this is an unprecedented situation due to the outbreak of COVID-19, we must practice social distancing. As a result, we did not have a in person team meeting. But we made a WhatsApp group named ‘DreamTeam’ where we continuously stayed in touch with each other. We discussed ideas and difficulties which we were facing and worked together as team to overcome those difficulties and come up with new ideas. Moreover, we used to have a group meeting daily on skype where we informed each other of the work done and about the work that was going to be done by a team member the following day. We used GitHub to collaborate.

**User Manual**

1. **User Session (Create and Read)**
   1. **Registration**

User can register himself to login in pro-dreamteam-registration.php.Screen of a cell phone

Description automatically generated

* 1. **Login**

Once user is registered, then he/she will be directed to login page.

1. **Professor’s Search, Ratings (Create, Read, Update and Delete)**
   1. **Search**

Once user is logged in, user can search for his/her professor’s name in the A group of people in a room

Description automatically generated

search bar. Autofill will work for the names already in database.

* 1. **Create**

After searching, user has the option to add the ratings for the Professor.

He can choose stars, course taken, and provide comments.

* 1. A screenshot of a cell phone

     Description automatically generated**Read**

User always reads the most updated version on the page.

* 1. **Update**

User has the option to edit the provided ratings.

* 1. **Delete**

Once the ratings are visible, logged in user has the option to delete

his/her ratings.

1. **Admin (CRUD)**
   1. A screenshot of a cell phone

      Description automatically generated**Read**

Admin can login using username: **admin** and password: **admin**. In the

admin page, data is read from database where admin can interact with

professor’s info.

* 1. **Create**

Only admin can add new professors.

* 1. **Update**

Admin can update the professor courses and assign more courses.

* 1. **Delete**

Admin can delete users, courses and professors.

1. A screenshot of a cell phone

   Description automatically generated**Laravel and CodeIgnitor**

Finally, use of two php frameworks to show the logout and about page of the website.

**Project Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Task | Start Date | End Date | % Done |
| 1 | 1) UI for Registration and Login Pages.  2) Developing entities and classes for students, reviews, instructors, and courses.  3)Developing database and tables for all the entities.  4) Finish Login and Registration pages back-end.  5) Session Handling. | 25/03/2020 | 1/04/2020 | 100% |
| 2 | 1) Implement the DAO classes for the entities.  2) Implement Professor Page back-end with all the CRUD features.  3) UI Design for all the pages  4) Input Validation.  5) Bug Fixing. | 1/04/2020 | 8/04/2020 | 100% |
| **3 Deadline** | 1) Implementation of web services within Professor Page using JSON.  2) Implementation of Admin Page with all CRUD operations for the entities.  3) Implementation of Laravel and CodeIgniter.  4) UI Designs and Modifications.  5) AutoFill for Professor search using JavaScript.  6) Debugging and Testing  7) Documentation Writing | 08/04/2020 | 14/04/2020 | 100% |

**Group Contributions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Components assigned | People/Person Assigned | Viacheslav  Sierkov | Dikshit Sharma | Robin  Singh Sidhu |
| Entities Creation | Everyone | ✔ | ✔ | ✔ |
| Admin Page front and back end | Robin |  |  | ✔ |
| PDOService Queries | Everyone | ✔ | ✔ | ✔ |
| Professor Page front and back end | Viacheslav, Dikshit | ✔ | ✔ |  |
| SQL Database Creation | Robin |  |  | ✔ |
| Login and Registration Page front and back end | Dikshit, Robin |  | ✔ | ✔ |
| Input Validation, Session Handling | Viacheslav | ✔ |  |  |
| Web Service Using JSON | Viacheslav, Dikshit | ✔ | ✔ |  |
| Laravel Integration | Everyone | ✔ | ✔ | ✔ |
| CodeIgniter Integration | Everyone | ✔ | ✔ | ✔ |
| Documentation Writing | Everyone | ✔ | ✔ | ✔ |

**Team Member Contribution Rate**

Robin Singh Sidhu – 30%

Viacheslav Sierkov – 30%

Dikshit Sharma – 40%

**References:**

1. *http://fontawesome.io*
2. *https://getbootstrap.com/*
3. *https://laravel.com/*
4. *https://codeigniter.com/*
5. *https://stackoverflow.com/*
6. *https://www.w3schools.com/php/*
7. *https://www.colorlib.com*
8. *http://dimsemenov.com/plugins/magnific-popup/*
9. *https://www.kunkalabs.com/mixitup/*
10. *http://creativecommons.org/licenses/by-nc/3.0/*
11. *https://colorlib.com/wp/template/login-form-v3/?v=3e8d115eb4b3*
12. *https://colorlib.com/wp/template/locals-directory/*