

# **NSP Report**

## **“CANDIDATE’S CATALOGUE”**

Submitted in  
partial fulfillment of the degree of Bachelor of Technology  
Rajasthan Technical University



By

**Prashant Sharma**  
**(PCE18CS115)**  
**Pawan Singh Gurjar**  
**(PCE18CS109)**  
**Naman Jain**  
**(PCE18CS097)**

Department of Computer Engineering  
Poornima College of Engineering, Jaipur  
(Academic Year 2021-22)

## **ABSTRACT**

An application of registry has been developed namely “Candidate’s Catalogue” using PHP , SQL,etc. This contains of 7 pages which are linked together using PHP, HTML,CSS etc. This application can be used in a company for storing the relevant information about an employee/applicant. Also, it canbe used in various education as well as charity institutions by making some minor changes in it.It stores the data of a user and does not erase it until it is deleted officially. This application contain the interlinking of eight different files which are being coded in several programming languages written above. Details of the components and elements used in making is described in this report.

## TABLE OF CONTENTS

<b>CHPATER 1: Introduction .....</b>	<b>11</b>
<b>1.1 Introduction.....</b>	<b>11</b>
<b>1.2 About the Company .....</b>	<b>11</b>
<b>1.3 Background of the Company.....</b>	<b>16</b>
<b>1.4 Coursera for Business .....</b>	<b>18</b>
<b>1.5 The Business Model .....</b>	<b>19</b>
<b>1.6 Future Plans .....</b>	<b>21</b>
<b>1.7 Conclusion .....</b>	<b>22</b>
<b>CHAPTER 2: Training Attended .....</b>	<b>15</b>
<b>2.1 Building Web Applications in PHP.....</b>	<b>23</b>
<b>2.2 Introduction to Structured Query Language (SQL).....</b>	<b>25</b>
<b>2.3 Building Database Applcation in PHP.....</b>	<b>27</b>
<b>2.4 Javascript,Jquery and JSON.....</b>	<b>28</b>
<b>2.5 Programming For Everybody.....</b>	<b>29</b>
<b>2.6 Python Data Structures.....</b>	<b>29</b>
<b>3.5 Conclusion.....</b>	<b>30</b>
<b>CHAPTER 3: System/Project Development.....</b>	<b>31</b>
<b>3.1 Project Description.....</b>	<b>31</b>
<b>3.2 Program Frames.....</b>	<b>32</b>
<b>3.3 Program Code.....</b>	<b>39</b>

<b>CHAPTER 4:</b>	<b>Conclusion .....</b>	<b>56</b>
<b>4.1</b>	<b>Conclusion.....</b>	<b>56</b>
<b>4.2</b>	<b>Future Scope .....</b>	<b>56</b>
<b>4.3</b>	<b>Refrences. ....</b>	<b>57</b>

## **LIST OF ABBREVIATIONS**

<b>Abbreviation</b>	<b>Acronym</b>
SQL	Structured Query Language
PHP	Hypertext Preprocessor
DOM	Document Object Model
JSON	JavaScript Object Notation
OO	Object Oriented
DOM	Document Object Model

## LIST OF FIGURES

Figure Number	Caption	Page No
Figure 1.	Main Frame .....	31
Figure 2.	Login Page.....	32
Figure 3.	Home Page.....	33
Figure 4.	Adding Profile page .....	34
Figure 5.	Profile Information .....	35
Figure 6.	Deleting Profile .....	36
Figure 7.	Editing Profile .....	37

## **1.1 Chapter 1: Introduction**

### **1.1 Introduction**

Coursera was founded by Daphne Koller and Andrew Ng with a vision of providing life-transforming learning experiences to anyone, anywhere. It is now a leading online learning platform for higher education, where 62 million learners from around the world come to learn skills of the future. More than 200 of the world's top universities and industry educators partner with Coursera to offer courses, Specializations, certificates, and degree programs. 2400 companies trust the company's enterprise platform Coursera for Business to transform their talent. Coursera for Government equips government employees and citizens with in-demand skills to build a competitive workforce. Coursera for Campus empowers any university to offer high-quality job-relevant online education to students, alumni, faculty, and staff. Coursera is backed by leading investors that include Kleiner Perkins, New Enterprise Associates, Learn Capital, and SEEK Group

### **1.2 About the Project**

Every course on Coursera is taught by top instructors from world-class universities and companies, so you can learn something new anytime, anywhere. Hundreds of free courses give you access to on-demand video lectures, homework exercises, and community discussion forums. Paid courses provide additional quizzes and projects as well as a shareable Course Certificate upon completion.

If you want to master a specific career skill, consider enrolling in a Specialization. You will complete a series of rigorous courses at your own pace, tackle hands-on projects based on real business challenges, and earn a Specialization Certificate to share with your professional network and potential employers

Whether you are looking to start a new career or change your current one, Professional Certificates on Coursera help you become job-ready. Learn at your own pace from top companies and universities, apply your new skills to hands-on projects that showcase your expertise to potential employers, unlock access to career support resources, and earn a career credential to kickstart your new career

With MasterTrack™ Certificates, portions of Master's programs have been split into online modules, so you can earn a high quality university-issued career credential at a breakthrough price in a flexible, interactive format. Benefit from a deeply engaging learning experience with real-world projects and live, expert instruction. If you are accepted to the full Master's program, your MasterTrack™ coursework counts towards your degree

Transform your career with a degree online from a world-class university on Coursera. Our modular degree learning experience gives you the ability to study on your own schedule and earn credit as you complete your course assignments. For a breakthrough price, you'll learn from top university instructors and graduate with an industry-relevant university credential.

## **1.3 Background**

### **1.3.1 History**

Coursera was founded in 2012 by Stanford University computer science professors Andrew Ng and Daphne Koller. Ng and Koller were inspired by their experiences offering their Stanford



courses online in fall 2011 and soon after left Stanford to launch Coursera. Princeton, Stanford, the University of Michigan and the University of Pennsylvania were the first universities to offer content on the platform. Offerings have since expanded to include Specializations – collections of courses that build skills in a specific subject – as well as degrees and a workforce development product for businesses and government organizations.

### **1.3.2 Funding**

The startup raised an initial \$16 million funding round backed by Kleiner Perkins Caufield & Byers and New Enterprise Associates. In 2013, the World Bank Group led the Series B investment, which totaled \$63 million. In 2015, EDB Investments led the Series C round of venture funding, which totaled more than \$60 million. In 2017, the company raised \$64 million from its existing investors in Series D round of funding. In 2019, the company raised \$103 million in Series E round of funding from the SEEK Group, Future Fund and NEA.<sup>[8]</sup> The company reached valuation of \$1 billion+ in 2019.

### **1.3.3 Strategic partners**

As of December 2019, the total number of partners is more than 200 across 29 countries. Coursera mainly works with universities and colleges, but also with corporate and governments. University partners include University of São Paulo in Brazil, University of London in the UK, Indian School of Business of India, Yonsei University in Korea, and institutions like Yale University of Illinois and University of Pennsylvania.

## 1.4 Business

Coursera was founded with the vision of providing life-transforming learning experiences to anyone, anywhere. We have come a long way since our launch in 2012; we now have over 21 million registered learners throughout the world, and we are bringing them outstanding educational content from 145 of the world's leading universities.

Today, we are taking yet another important step in our effort to expand the Coursera learner community. I am excited to announce Coursera for Business, our enterprise platform for workforce development at scale. We see Coursera for Business as a natural extension of our vision, and as a powerful way to help leading companies around the world address the rapidly evolving training and development needs of their employees.

Since Coursera's inception, we have kept learners at the center of our approach as we consider new growth opportunities. We've always known that many of our learners are housed within companies, and that many of you are using Coursera to build skills relevant to your jobs. Over the past four years, Coursera Certificates have become the second most-cited credential on LinkedIn; at the same time, our conversations with corporate leaders have highlighted the content, cost, and scale challenges of providing high-quality training and development opportunities to a distributed workforce. These trends and insights have inspired us to develop a solution that enables your employers to build our courses and Specializations – offered by the world's best universities and education institutions – into the fabric of their development programs that are meant to help you improve and grow at those companies.

The best employees increasingly care about development, and development opportunities are one of the top reasons that millennials choose to work and stay at a company. Coursera for Business will help companies give these lifelong learners access to curated learning programs that draw on over 1,400 university courses, delivered through a robust online learning experience on both desktop and mobile.

We are privileged to be working toward our vision alongside 145 university partners and a truly global community of learners. Today's launch is one big step toward engaging another major stakeholder – leading employers that are committed to giving their employees access to the world's best learning and development experiences.

## **1.5 The Business Model**

People with utopian visions for the internet first created mOOCs. This means the idea for platforms like Coursera was likely conceived without a business plan in mind. Nonetheless, over the past seven years, Coursera has managed to monetize its platform. It is worth noting, however, that monetization has led to the effective elimination of the original MOOC idea, which is predicated on ideals like free and open access, as well as the building of online communities. Today, Coursera users must pay to engage with material in a meaningful way and take courses for individualistic purposes. This has been a consistent trend among all major online education platforms.

Coursera has achieved a stable business model by offering products at a wide range of prices. These include over 3,100 courses that are free to audit, “Signature Tracks” for \$30 to \$100, “Specializations” for \$39 to \$89 monthly subscriptions, online degree programs for \$15,000 to \$30,000, and “Coursera for Business” for \$400 per employee per year.

### **1.5.1 Fundraising**

Coursera has raised over \$313.1 million over nine rounds of fundraising. On April 25, the company secured a \$103 million investment in a round lead by SEEK Group, an online employment marketplace and a global leader in investing, scaling, and operating online

employment and education businesses. This latest round has increased speculation among industry experts that Coursera might be nearing an IPO.

### **1.5.2 Signature Track**

The lectures for the majority of the courses Coursera offers are free. However, students who wish to earn a certificate of completion can opt to pay a per-course fee to participate in the "Signature Track." This feature, which Coursera launched in January 2013 and that became a platform-wide norm in 2015, gives paying students access to graded assignments, homework, and examinations. If completed in a satisfactory manner, Signature Track students earn a verified certificate at the end of the course, which is emblazoned with the name of the course and the university that provided its content. Students can then show these certificates to employers to prove their professional qualifications and acquired skills. They cost between \$30 and \$100, depending on the course. Coursera also offers financial aid to students who can demonstrate need, although the requirements for such aid have become increasingly stringent as the company has grown.

These certificates, which afford the platform legitimacy in the eyes of employers, have become the backbone of Coursera's business. Coursera's other business sectors, like "Specializations" and degree programs, also use certificates.

The Signature Track was the Coursera's first successful attempt to monetize its platform. Just over a year after launching the Signature Track, Coursera had generated more than \$4 million in cumulative revenues from it.

### **1.5.3 Specializations**

A year after it launched the Signature Track, Coursera launched "Specializations." This feature is essentially a bundle of different courses designed to help students deepen their expertise in a certain area; to specialize, if you will. Students pay monthly subscription fees of between \$39

and \$89, depending on the Specialization, for access to graded assignments and the eventual certificate.

#### **1.5.4 Degree Students**

Finally, Coursera offers students the possibility to earn fully accredited degrees on its platform. Coursera offers 14-degree programs from top universities, including popular programs like MBAs, machine-learning programs, and computer science programs. In Jan. 2018, Coursera announced 1,632 students were enrolled in degree programs and that the company had earned \$9.8 million in revenue from said students. These programs cost between \$15,000 and \$30,000.

### **1.6 Future Plans**

#### **1.6.1 Backing Up with High Revenue**

In order to secure more rounds of investment, Coursera will have to demonstrate predictably high revenue growth going forward. New features and a quickly growing user base is one thing, but the ability to monetize these things effectively is another. Currently, only 40% of paying students finish the courses they enroll in. Coursera needs to find ways to boost this number, because more completed courses means more additional courses purchased. According to Dil Sidhu, the company's Chief Content Officer, Coursera is planning on employing more data analytics tools (read: AI) to study why students don't complete courses and adding behavioral sciences lessons to coach students to be more disciplined.

During its 2019 partner's conference, several panelists also urged those offering courses to "break the fourth wall" by adding more personal touches to lessons. The thinking here is that by making online education a more accurate simulation of classroom learning, it will become more engaging. To the same aim, speakers spoke about the need for more interaction between

teachers and students. This would mean that universities would have to make more professors available to interface with online students.

### **1.6.2 Partnering with Governments and Big Business**

Coursera's deal with the Abu Dhabi School of Government, as mentioned above, is representative of another growing trend in Coursera's business: partnering with governments and businesses with massive workforces. Coursera also currently works with Singapore, Egypt, and India. In a conversation with Forbes, Coursera's CEO, Jeff Maggioncalda, cited a McKinsey report that warns of 50% of today's jobs being at risk of replacement by automation. Coursera plans to capitalize on a product of this trend: increased demand for retraining and lifelong education. And the company seems to be doing just that. It already services over 60 Fortune 500 companies, including Amazon (AMZN), Google (GOOGL), and P&G (PG). Moreover, as automation continues to increase the demand for retraining, this number will likely rise.

## **1.7 Conclusion**

Coursera Inc. is an online education provider that offers students access to massive open online courses (MOOCs), specializations, and even degrees. Founded in 2012 by Stanford computer science professors Andrew Ng and Daphne Koller, Coursera does not actually create educational content itself. Rather, the company partners with universities and other organizations to provide them with an online platform that students pay to gain access to.

## Chapter 2

### **Training Attended**

#### **2.1 Building Web Applications in PHP**

In this course, i have explored the basic structure of a web application, and how a web browser interacts with a web server. I was introduced to the request/response cycle, including GET/POST/Redirect. I have also gained an introductory understanding of Hypertext Markup Language (HTML), as well as the basic syntax and data structures of the PHP language, variables, logic, iteration, arrays, error handling, and superglobal variables, among other elements. An introduction to Cascading Style Sheets (CSS) will allow you to style markup for webpages. Lastly, I gained the skills and knowledge to install and use an integrated PHP/MySQL environment like XAMPP or MAMP

##### **2.1.1 PHP**

PHP is a popular general-purpose scripting language that is especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of a HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. Arbitrary PHP code can also be interpreted and executed via command-line interface (CLI)

### Data Types

PHP stores integers in a platform-dependent range, either as a 32, 64 or 128-bit signed integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations, which is different behavior to many other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, hexadecimal, and binary notations.

Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++.

### Objects

PHP 5 introduced private and protected member variables and methods, along with abstract classes, final classes, abstract methods, and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model. Furthermore, PHP 5 added interfaces and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. Objects implementing Array Access can be used with array syntax and objects implementing Iterator or Iterator Aggregate can be used with the for each language construct. There is no virtual table feature in the engine, so static variables are bound with a name instead of a reference at compile time.

#### **2.1.2 MAMP**

MAMP is a solution stack composed of free and open-source and proprietary commercial software used together to develop and run dynamic web sites on Apple Macintosh computers.



The name MAMP is an acronym that stems from the names of the components of the system macOS: Apache (the web server); MySQL or MariaDB (the database management system); and PHP, Perl, or Python (programming languages used for web development). The name is derived from LAMP, a similar stack of all open-source software widely used for web sites, but substituting the proprietary macOS for the open-source Linux OS. (Similar "AMP" stacks exist for other operating systems.) MAMP is not limited to these choices of components, however; Nginx can be used in place of Apache, for example, and the same goes for substituting MariaDB for MySQL.

Some of the software packages that comprise MAMP (particularly Apache and PHP) are pre-installed with macOS; compatible versions of the remainder are readily available for installation and use. MAMP is commonly used with and to develop for popular CMS programs such as WordPress and Drupal by setting up a local development environment on laptop or desktop computers, without the need for a standalone web server.

## **2.2 Introduction to Structured Query Language (SQL)**

In this course, we walk through installation steps for installing a text editor, installing MAMP or XAMPP (or equivalent) and creating a MySQL Database. I also learnt about single table queries and the basic syntax of the SQL language, as well as database design with multiple tables, foreign keys, and the JOIN operation. Lastly, also learnt to model many-to-many relationships like those needed to represent users, roles, and courses.

### **2.2.1 Structured Query Language (SQL)**

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify *how* to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), [b] a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements.

### **2.2.2 Relational Database**

A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

### **2.2.3 MySQL**

MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database.

## **2.3 Building Database Applications in PHP**

In this course, I looked at the object oriented patterns available in PHP. I also learned how to connect to a MySQL using the Portable Data Objects (PDO) library and issue SQL commands in the the PHP language. I also looked at how PHP uses cookies and manages session data. I also learnt how PHP avoids double posting data, how flash messages are implemented, and how to use a session to log in users in web applications. Then build the first 'complete' application that has multiple screens to Create, Read, Update and Delete (CRUD) our data. This brings all the previous concepts together and will form the basis for all later web applications.

### **2.3.1 Database Application**

A database application is a computer program whose primary purpose is entering and retrieving information from a computerized database. Early examples of database applications were accounting systems and airline reservations systems, such as SABRE, developed starting in 1957.

A characteristic of modern database applications is that they facilitate simultaneous updates and queries from multiple users. Systems in the 1970s might have accomplished this by having each user in front of a 3270 terminal to a mainframe computer. By the mid-1980s it was becoming more common to give each user a personal computer and have a program running on that PC that is connected to a database server. Information would be pulled from the database, transmitted over a network, and then arranged, graphed, or otherwise formatted by the program running on the PC. Starting in the mid-1990s it became more common to build database applications with a Web interface. Rather than develop custom software to run on a user's PC, the user would use the same Web browser program for every application. A database application with a Web interface had the advantage that it could be used on devices of different sizes, with different hardware, and with different operating systems. Examples of early database applications with Web interfaces include amazon.com, which used the Oracle relational database management system, the

photo.net online community, whose implementation on top of Oracle was described in the book Database-Backed Web Sites (Ziff-Davis Press; May 1997), and eBay, also running Oracle.

## **2.4 JavaScript, jQuery, and JSON**

In this course, I looked at the JavaScript language, and how it supports the Object-Oriented pattern, with a focus on the unique aspect of how JavaScript approaches OO. Also, I explored a brief introduction to the jQuery library, which is widely used to do in-browser manipulation of the Document Object Model (DOM) and event handling. I also learnt more about JavaScript Object Notation (JSON), which is commonly used as a syntax to exchange data between code running on the server (i.e. in PHP) and code running in the browser (JavaScript/jQuery).

### **2.4.1 jQuery**

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having 3 to 4 times more usage than any other JavaScript library

### **2.4.2 Javascript**

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The

vast majority of websites use it for client-side page behavior.and all major web browsers have a dedicated JavaScript engine to execute it.

## **2.5 Programming for Everybody (Getting Started with Python)**

This course aims to teach everyone the basics of programming computers using Python. It covers the basics of how one constructs a program from a series of simple instructions in Python. The course has no pre-requisites and avoids all but the simplest mathematics. Anyone with moderate computer experience should be able to master the materials in this course.

### **2.5.1 Python**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

## **2.6 Python Data Structures**

This course will introduce the core data structures of the Python programming language. We will move past the basics of procedural programming and explore how we can use the Python built-in

data structures such as lists, dictionaries, and tuples to perform increasingly complex data analysis.

### **2.6.1 Data Structures in Python**

Python has been used worldwide for different fields such as making websites, artificial intelligence and much more. But to make all of this possible, data plays a very important role which means that this data should be stored efficiently and the access to it must be timely. So how do you achieve this? We use something called Data Structures. With that being said, let us go through the topics we will cover in Data Structures in Python.

Organizing, managing and storing data is important as it enables easier access and efficient modifications. Data Structures allows you to organize your data in such a way that enables you to store collections of data, relate them and perform operations on them accordingly.

## **2.6 Conclusion**

In the duration of 45 days, the criteria of 120 hours has been fulfilled by me. During this period I completed one specialization named 'Web Application for Everybody' which included four different courses. All these certificates have been attached above as per the format. Two courses of python were also completed. Four out of six courses were of intermediate level and remaining two were of beginner level. All the six courses belongs to University of Michigan and were instructed by Charles Serverance. These courses were very interesting as well as vital as it helped me in gaining some new skills.

## **Chapter 3**

### **Systems/Project Development**

#### **3.1 Project Description**

##### **3.1.1 Introduction/ About the Project**

As is the case with all professions, the key to landing a job carefully crafted, shining resume. And when it comes to building this essential document, you can either create your resume using templates and resume-building software tools, or you can draft, edit and format your own document from the ground up.

No matter which option you choose, LiveCareer can help. Even if you decide to create your own application, take a few minutes to review this collection of registrar resume samples. These samples can serve as a model and guide, and they can help you understand what your employers will be looking for in terms of language, structure, layout, and necessary information.

In this project I have used PHP, HTML and CSS for creating a web application of Candidate's Catalogue. This project is very helpful in storing the basic information about an applicant of a particular company. Authorised person can edit the information/data of someone. If someone has left the company then his/her data can also be deleted by directly clicking on the delete button corresponding to his/her name. Full information of any employee can be seen by clicking on the name directly. Only limited number of people can access the data stored on the server. Employee whose e-mail id has been entered in the database can only access this data.

### 3.2 Program Frames

The main frame of the program looks like this

## Prashant Sharma's Resume Registry

Please log in

Name	Grade
Prashant Sharma	8.65
Rajesh Arora	8.51

**Note:** Your implementation should retain data across multiple logout/login sessions. This sample implementation clears all its data periodically implementation.

Main frame contains ‘Please log in’ button. And as their name suggests they perform action as desired. On verifying authentication details, the user gets an option to enter the system of registry..



## Please Log In

Email

Password

For a password hint, view source and find a password hint in the HTML comments.

This is a login page. We can easily login into this system by entering the valid e-mail id as well as password. There are two buttons “Log In” and “Cancel” which perform the same function as their name suggests.

# Prashant Sharma's Resume Registry

[Logout](#)

Name	Grade	Action
<a href="#">Prashant Sharma</a>	8.65	<a href="#">Edit</a> / <a href="#">Delete</a>

[Add New Entry](#)

**Note:** Your implementation should retain data across multiple logout/login sessions. This sample implementation clears all its data on logout - with your implementation.

This is a main page which contains different links as “Logout”, “Add New Entry”. Clicking on any of the names will take you to the different page displaying whole information about a person.

We can also edit and delete the information about a particular person.

## Adding Profile

First Name:

Last Name:

Email:

Grade in UG/PG:

Experience/ Achievement:  

Data Science, ML

This frame consists of lot of entries, which contains the information about a person's registry. It is compulsory to fill all the details otherwise it will show an error. Also data type of the entries should be entered accordingly. "Add" button will add the entry to the main data available.

# Profile information

First Name: Prashant

Last Name: Sharma

Email: 2018pcecsprashant115@poornima.org

Grade:

8.65

Experience/ Achievement:

I have learnt few programming languages like PHP, Python, C++ etc. Also, I am the captain of Excellence Cricket Club of Poornima College

[Done](#)

By clicking on the name whole information about a person will be displayed in this format.

## Deleting Profile

First Name: Prashant

Second Name: Sharma

Delete

Cancel

When u will click on the delete button on main page then firstly, it will display a confirmation message then it will take the final action of deleting.

## Editing Profile

First Name:

Last Name:

Email:

Grade:

Experience/ Achievement:

Know Java, C++, Python etc.  
Worked as a member of drama group "Jazba".  
Communication Skills.

After clicking on the edit button it will redirect us at the above page where we can edit the information which we want and can save it easily by clicking on the Save button.

### 3.3 Complete code

#### 3.3.1 Code of add.php

```
<?php
```

```
session_start();
```

```
$pdo = new PDO('mysql:host=localhost;port=3306;dbname=misc', 'fred', 'zap');
```

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
if (!isset($_SESSION['login'])) {
```

```
    die("ACCESS DENIED");
```

```
} else if (isset($_POST['cancel'])) {
```

```
    header('Location: index.php');
```

```
    return ;
```

```
} else {
```

```
    if (isset($_POST['first_name']) and isset($_POST['last_name']) and isset($_POST['email'])  
and isset($_POST['headline']) and isset($_POST['summary'])) {
```

```
        $firstName = $_POST['first_name'];
```

```
        $lastName = $_POST['last_name'];
```

```
        $email = $_POST['email'];
```

```
        $headline = $_POST['headline'];
```

```
        $summary = $_POST['summary'];
```

```
        if ($firstName == "" or $lastName == "" or $email == "" or $headline == "" or $summary  
== "") {
```

```
$_SESSION['errors'] = "All fields are required";

header("Location: add.php");

return ;

} else {

    if (strpos($email, '@') == false) {

        $_SESSION['errors'] = "Email address must contain @";

        header("Location: add.php");

        return ;

    } else {

        $stmt = $pdo->prepare('INSERT INTO Profile

                                (user_id, first_name, last_name, email, headline, summary) VALUES

                                (:userId, :firstName, :lastName, :email, :headline, :summary)');

        $stmt->execute(array(':userId' => $_SESSION['user_id'],

                                ':firstName' => $firstName,

                                ':lastName' => $lastName,

                                ':email' => $email,

                                ':headline' => $headline,

                                ':summary' => $summary)

                                );

        $_SESSION['success'] = "Profile added.";

        header("Location: index.php");
```



```
        return ;  
    }  
}  
}  
}  
}  
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Prashant Sharma</title>
```

```
<link rel="stylesheet"
```

```
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
```

```
    integrity="sha384-
```

```
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
```

```
    crossorigin="anonymous">
```

```
<link rel="stylesheet"
```

```
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css"
```

```
    integrity="sha384-
```

```
fLW2N01lMqjakBkx3l/M9EahuwPsFeNvV63J5ezn3uZzapT0u7EYsXMjQV+0En5r"
```

```
    crossorigin="anonymous">
```

```
<link rel="stylesheet"
      href="https://code.jquery.com/ui/1.12.1/themes/ui-lightness/jquery-ui.css">
```

```
<script
src="https://code.jquery.com/jquery-3.2.1.js"
integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jcIzLy3oi35EouyE="
crossorigin="anonymous"></script>
```

```
<script
src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"
integrity="sha256-T0Vest3yCU7pafRw9r+settMBX6JkKN06dqBnpQ8d30="
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<h1>Adding Profile for
```

```
<?php
```

```
    echo htmlentities($_SESSION['name']);
```

```
?>
```

</h1>

<div style="color: red;">

<?php

    echo \$\_SESSION['errors'];

    unset(\$\_SESSION['errors'])

?>

</div>

<form method="post">

    <p>First Name:

        <input type="text" name="first\_name" size="60"/>

    </p>

    <p>Last Name:

        <input type="text" name="last\_name" size="60"/>

    </p>

    <p>

        Email:

        <input type="text" name="email" size="30"/>

    </p>

```
<p>
    Headline:<br/>
    <input type="text" name="headline" size="80"/>
</p>

<p>
    Summary:<br/>
    <textarea name="summary" rows="8" cols="80"></textarea>
<p>
    <input type="submit" value="Add">
    <input type="submit" name="cancel" value="Cancel">
</p>
</form>
</div>

<script    data-cfasync="false"    src="/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-
decode.min.js"></script>

</body>
</html>
```

### 3.3.2 Code of index.php

```
<?php
```

```
    session_start();
```

```
    $pdo = new PDO('mysql:host=localhost;port=3306;dbname=misc', 'fred', 'zap');
```

```
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
    $stmt = $pdo->query("SELECT profile_id, first_name, last_name, email, headline, summary  
                        FROM Profile");
```

```
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

```
?>
```

```
<html>
```

```
    <head>
```

```
        <title>Prashant Sharma</title>
```

```
        <link rel="stylesheet"
```

```
            href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
```

```
            integrity="sha384-
```

```
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
```

```
            crossorigin="anonymous">
```

```
        <link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css"
integrity="sha384-
fLW2N01lMqjakBkx3l/M9EahuwPsfENvV63J5ezn3uZzapT0u7EYsXMjQV+0En5r"
crossorigin="anonymous">

<link rel="stylesheet"
href="https://code.jquery.com/ui/1.12.1/themes/ui-lightness/jquery-ui.css">

<script
src="https://code.jquery.com/jquery-3.2.1.js"
integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jclLy3oi35EouyE="
crossorigin="anonymous"></script>

<script
src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"
integrity="sha256-T0Vest3yCU7pafRw9r+settMBX6JkKN06dqBnpQ8d30="
crossorigin="anonymous"></script>

</head>

<body>

<div class="container">

<h2>Prashant Sharma's Candidate's Catalogue</h2>
```

```
<?php
    if (!$_SESSION['login']) {
        echo '
        <p><a href="login.php">Please log in</a></p>';

    if ($rows == false) {

    } else {
        echo "<table border='1'>
            <thead>
            <tr>
                <th>Name</th>
                <th>Headline</th>
            </tr>
            </thead>";

        foreach ($rows as $row) {
            echo "<tr><td>";

            $full_profile_name = $row['first_name'] . " " . $row['last_name'];

            echo('<a href="view.php?profile_id=' . $row['profile_id'] . "'> .
htmlentities($full_profile_name) . '</a>');
        }
    }
}
```

```
        echo("</td><td>");  
  
        echo(htmlentities($row['headline']));  
  
        echo("</td></tr>\n");  
    }  
    echo "</table>";  
}
```

```
echo '<p>
```

**Note:** Your implementation should retain data across multiple logout/login sessions.

This sample implementation clears all its data periodically - which you should not do in your implementation.

```
</p>';
```

```
exit();
```

```
}
```

```
?>
```

```
<p>
```

```
<a href="logout.php">Logout</a>
```

```
</p>
```



```
<?php

if ($rows == false) {

} else {

    echo "<table border='1'>

        <thead>

            <tr>

                <th>Name</th>

                <th>Headline</th>

                <th>Action</th>

            </tr>

        </thead>";

    foreach ($rows as $row) {

        echo "<tr><td>";

        $full_profile_name = $row['first_name'] . " " . $row['last_name'];

        echo('<a href="view.php?profile_id=' . $row['profile_id'] . "'>';

        htmlentities($full_profile_name) . '</a>');

        echo("</td><td>");

        echo(htmlentities($row['headline']));

        echo("</td><td>");
```

```
        echo('<a href="edit.php?profile_id=' . $row['profile_id'] . "'>Edit</a> / ');
        echo('<a href="delete.php?profile_id=' . $row['profile_id'] . "'>Delete</a>');
        echo("&</td></tr>\n");
    }
    echo "</table>";
}
?>

<p>
    <a href="add.php">Add New Entry</a>
</p>

<p>
    <b>Note:</b> Your implementation should retain data across multiple
    logout/login sessions. This sample implementation clears all its
    data on logout - which you should not do in your implementation.
</p>
</div>
</body>
</html>
```

### 3.3.3 Code of login.php

```
<?php
```

```
session_start();
```

```
$pdo = new PDO('mysql:host=localhost;port=3306;dbname=misc', 'fred', 'zap');
```

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
if (isset($_POST['cancel'])) {
```

```
    header("Location: index.php");
```

```
    return ;
```

```
}
```

```
if (isset($_POST['email']) and isset($_POST['pass'])) {
```

```
    $user = $_POST['email'];
```

```
    $pass = $_POST['pass'];
```

```
    $salt = "XyZzy12*_";
```

```
    $check = hash('md5', $salt.$_POST['pass']);
```

```
    $stmt = $pdo->prepare('SELECT user_id, name
```

```
        FROM users
```

```
        WHERE email = :em AND password = :pw');
```

```
    $stmt->execute(array(':em' => $_POST['email'], ':pw' => $check));
```

```
$row = $stmt->fetch(PDO::FETCH_ASSOC);

if ($row == false) {
    $_SESSION["errors"] = "Incorrect Password \n";
    error_log("Login fail " .$_POST['name']. $actualHash);
    header("Location: login.php");
    return ;
} else {
    error_log("Login success ".$user);
    $_SESSION['name'] = $row['name'];
    $_SESSION['login'] = true;
    $_SESSION['user_id'] = $row['user_id'];
    header("Location: index.php");
    return ;
}
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
integrity="sha384-
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
crossorigin="anonymous">
```

```
<!-- Optional theme -->
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css"
integrity="sha384-
fLW2N01lMqjakBkx3l/M9EahuwpsFeNvV63J5ezn3uZzapT0u7EYsXMjQV+0En5r"
crossorigin="anonymous">
```

```
<!-- Custom styles for this template -->
```

```
<link href="starter-template.css" rel="stylesheet">
```

```
<title>Prashant Sharma</title>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<h1>Please Log In</h1>
```

```
<div style="color: red;">
```

```
<?php

    //echo $_SESSION["errors"];

    unset($_SESSION["errors"]);

?>

</div>


<form method="POST">

    <label for="email">Email</label>

    <input type="text" name="email" id="email"/><br/>

    <label for="id_1723">Password</label>

    <input type="password" name="pass" id="id_1723"/><br/>

    <input type="submit" onclick="return doValidate();" value="Log In">

    <input type="submit" name = "cancel" value="Cancel">

</form>


<p>

    For a password hint, view source and find a password hint

    in the HTML comments.

    <!-- Hint: The password is the three character name of the

    programming language used in this class (all lower case)

    followed by 123. -->

</p>
```

```
<script>

function doValidate() {

    console.log('Validating...');

    try {

        addr = document.getElementById('email').value;

        pw = document.getElementById('id_1723').value;

        console.log("Validating addr="+addr+" pw="+pw);

        if (addr == null || addr == "" || pw == null || pw == "") {

            alert("Both fields must be filled out");

            return false;

        }

        if ( addr.indexOf('@') == -1 ) {

            alert("Invalid email address");

            return false;

        }

        return true;

    } catch(e) {

        return false;

    }

    return false;

}
```

```
</script>

</div>

</body>

</html>
```

## **Chapter 4**

### **Conclusion**

#### **4.1 Conclusion**

This project is very helpful in storing the basic information about an employee of a particular company. One can edit the information/data of someone. If someone has left the company then his/her data can also be deleted by directly clicking on the delete button corresponding to his/her name. Full information of any employee can be seen by clicking on the name directly. Only limited number of people can access the data stored on the server. Employee whose e-mail id has been entered in the database can only access this data. Coursera helped me to complete my summer training in a perfect manner by providing high level programs and courses efficiently.

#### **4.2 Future scope**

I would improve the performance of the project by especially adding some more features to the different pages. Also, I will try to create backup of the data stored on the web server. SQL injection is also a good option to perform on it. More pages can be linked with a single page so that we can store the detailed information about an employee. It can also be linked with the annual performance of an employee.

Other advanced implementation include the different retrieving techniques so that the data lost can be recovered easily. Also, the security of the data is also very important as the personal data of an employee should not be disclosed. Hence, we can increase the security patch on it.



### 4.3 REFERENCES

[www.wikipedia.org](http://www.wikipedia.org)

[www.coursera.org](http://www.coursera.org)

[www.geeksforgeeks.org](http://www.geeksforgeeks.org)

[www.edureka.co](http://www.edureka.co)

[en.wikipedia.org](http://en.wikipedia.org)

[www.mysql.com](http://www.mysql.com)

[www.wa4ae.com](http://www.wa4ae.com)

