## CHAPTER 1

**INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

Internship process for students is web application it helps to process internship work through online. An internship experience combines hands-on workplace experience with your discipline-specific skills, and earns academic credit towards your course. Arranging your internship involves three steps: preparing, securing, and registering. An internship experience combines hands-on workplace experience with your discipline-specific skills, and earns academic credit towards your course. Arranging your internship involves three steps: preparing, securing, and registering.

As the Internship in Online accessibility software, **“INTERNSHIP PROCESS ADMINISTRATION SYSTEM”** has helped students to view the company’s availability of vacancies in internship, how to apply the procedure and techniques details are easily track through this website process, and also they check of interviews available for internship, company can post their requirement.

This project is aimed at developing by using JAVA as Front end and MySQL Server as Back end.

## CHAPTER 2

**SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM**

In existing system there is an no proper way to communicate with client and the internship students. Students they need to search the job or intern by themselves. Which is more complex to find the opportunity to prove our knowledge

## Drawbacks of Existing System

The following are the drawbacks of the existing system

* + - Complex to managing the internship process.
    - Due to that they face problems like data loss.
    - Can’t follow the up-to-date status of the project.

## PROPOSED SYSTEM

This new interface shall enable the students to apply for internship. To use this system, the students should have registered with the database. Each student shall have a login id and password. The admin shall have the rights to create and manage these users from the back end. When the students log in to the system, a grid view shall be displayed which displays all the open job positions. It also gives the details about the start and end date. It gives a brief description about the job. Using the apply button the student shall be able to apply for the job.

## Advantages of Proposed System

The following are the advantage of proposed system

* + - Very easy to understand the flow of project structure.
    - Could be gave more opportunity to the student
    - Easy to track the intern students

## FEASIBILITY STUDY

The feasibility of the system is analyzed in this phase and business proposal is put forth with general plan for the project and cost estimates. During the system analysis of the project, the feasibility study of proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in feasibility analysis are

* + - Technical Feasibility
    - Economic Feasibility
    - Operational Feasibility

## Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, whether the certain current resources and technology can be upgraded or added in the software to accomplish specified user requirements.

The technical requirements of the application are simple and basic. Python is used for the developers of the application and the framework is largely used by many, thus there will be enough support for future enhancements. The framework is stable and the support from the developers is constantly updated. The devices which have internet connectivity are enough for the application.

## Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on.

The cost of application development is very less and the cost of implementation is also less. It can be developed with the system with minimum requirements and can also be operated with the system with some basic requirements that are available the existing systems. For this, it is essential to consider expenses made on purchases and activities required to carry out software development.

## Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve user requirements. This feasibility is dependent on developer and involves visualizing whether the software will operate after it has been developed and be operative.

The application is developed based on the user requirements and is developed on the priority of the user requirements such as an integrated service and reviewing platform.

## PROBLEM ANALYSIS

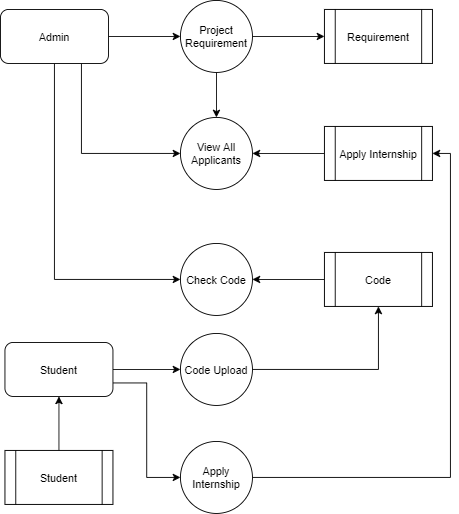
The orders from public are submitted through email, letter or form to the corresponding authority. The registered orders are forwarded to corresponding authority and report about the orders is submitted. The status of orders addressed is not recorded and maintained properly. The maps are not implemented. So, the customer does not know live location of the package. There is need for the software which receives the orders through online, forwarded to respective peoples for rectification and post the status of orders.

## CONTEXT AND DATA FLOW DIAGRAM

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or system. The DFD also provides information about the outputs and inputs of each entity and process itself. A data-flow diagram is a part of structured-analysis modelling tools.



**LEVEL 1:**



* 1. **SYSTEM CONFIGURATION**

### Hardware Requirements

Processor : Intel Core i5 RAM Capacity : 4 GB

Hard Disk : 120 GB

### Software Requirements

Operating System: Windows 8,10 Ultimate Front End : HTML AND CSS

Server used : MYSQL

Back End : JAVA

### Software Descriptions HTML

Hypertext Markup Language (HTML) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) it forms a triad of cornerstone technologies for the [World Wide Web.](https://en.wikipedia.org/wiki/World_Wide_Web) [Web](https://en.wikipedia.org/wiki/Web_browser) [browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and render them into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects, such as [interactive forms,](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. It provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links,](https://en.wikipedia.org/wiki/Hyperlink) and other items.

HTML elements are delineated by *tags*, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as,

<img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript) which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. These are designed to be usable by all Open Web developers, this reference page links to numerous resources about HTML5 technologies, classified into several groups based on their function.

### CSS

Cascading Style Sheets (CSS) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [mark-up language](https://en.wikipedia.org/wiki/Markup_language). Although most often used to set the visual style of [web pages](https://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](https://en.wikipedia.org/wiki/HTML) and [XHTML,](https://en.wikipedia.org/wiki/XHTML) the language can be applied to any [XML](https://en.wikipedia.org/wiki/XML) document, including [plain XML,](https://en.wikipedia.org/wiki/Plain_Old_XML) [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL,](https://en.wikipedia.org/wiki/XUL) and is applicable to rendering in [speech,](https://en.wikipedia.org/wiki/Speech_synthesis) or on other media. Along with HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for [web applications](https://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility,](https://en.wikipedia.org/wiki/Accessibility) provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS.

### Java

Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.

The Java syntax is similar to C++, but is strictly an object-oriented programming language. For example, most Java programs contain classes, which are used to define objects, and methods, which are assigned to individual classes. Java is also known for being more strict than C++, meaning variables and functions must be explicitly defined. This means Java source code may produce errors or "exceptions" more easily than other languages, but it also limits other types of errors that may be caused by undefined variables or unassigned types.

Unlike Windows executables (.EXE files) or Macintosh applications (.APP files), Java programs are not run directly by the operating system. Instead, Java programs are interpreted by the Java Virtual Machine, or JVM, which runs on multiple platforms. This means all Java programs are multiplatform and can run on different platforms, including Macintosh, Windows, and Unix computers. However, the JVM must be installed for Java applications or applets to run at all. Fortunately, the JVM is included as part of the Java Runtime Environment (JRE),

### MySQL

MySQL is the popular Open Source Relational SQL Database Management System. MySQL is being used for developing various web-based software applications. The MySQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms of the [General Public](https://en.wikipedia.org/wiki/GNU_General_Public_License) [License.](https://en.wikipedia.org/wiki/GNU_General_Public_License) It is used to store the information.

MySQL was owned and sponsored by the single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

## CHAPTER 3

**SYSTEM DESIGN**

**3.1 INPUT DESIGN**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. Input Design is the first phase in the system design. Input designing is to converting the user-oriented information to the computer- oriented form. The input data items are grouped and analyzed to find out whether the proposed system can be developed from the user input. The system is developed using various processes screens formats.

The main objective of input design is to

* + - It should serve specific purpose effectively such as storing, recording, and retrieving the information.
    - It ensures proper completion with accuracy.
    - It should be easy to fill and straightforward.
    - It should focus on user’s attention, consistency and simplicity.

### DATABASE DESIGN

The most important consideration in designing the database is how the information will be used. The main objective of designing a database is Data Integration, Data Integrity and Data Independence.

### Data Integration

In a database, information from several files are coordinated, accessed and operated upon as through it is in a single file. Logically, the information is centralized, physically, the data may be located on different devices, connected through data communication facilities.

### Data Integrity

Data integrity means storing all data in one place only and how each application access it. This approach results in more consistent information, one update being sufficient to achieve a new record status for all applications. This leads to less data redundancy, that is data items need not be duplicated. A reduction in the direct access storage requirement.

### Data Independence

Data independence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of application and allow modifications to application programs without reorganizing the physical data.

### 3.2.1 TABLE DESIGN

The table needed for each module were designed and the specification of each and every column was given based on the records and details collected during record specification of the system study.

### Table 3.1 Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD TYPE** | **SIZE** | **CONSTRAINT** |
| Id | Int | 10 | PRIMARY KEY |
| Username | Varchar | 15 | Not null |
| Password | Varchar | 20 | Not null |

**Table 3.2 Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD TYPE** | **SIZE** | **CONSTRAINT** |
| Id | Int | 10 | Primary key |
| Applicant\_name | Varchar | 15 | Not null |
| Qualification | Varchar | 15 | Not null |
| Mobile | Varchar | 15 | Not null |
| Front end | Varchar | 15 | Not null |
| Back end | Varchar | 15 | Not null |
| Database | Varchar | 15 | Not null |
| Designation | Varchar | 15 | Not null |
| Username | Varchar | 15 | Not null |
| Password | Varchar | 15 | Not null |

**Table 3.3 requirement**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD TYPE** | **SIZE** | **CONSTRAINT** |
| Id | Int | 10 | Primary key |
| Title | Varchar | 30 | Not null |
| Description | Varchar | 100 | Not null |
| Front end | Varchar | 15 | Not null |
| Back end | Date | 20 | Not null |
| Modules | Varchar | 100 | Not null |

**Table 3.4 intern**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD TYPE** | **SIZE** | **CONSTRAINT** |
| Id | Int | 10 | Primary key |
| Project id | Int | 10 | Foreign key |
| Student id | Int | 10 | Foreign key |
| Code | Varchar | 10000 | Not null |
| date | Date | 10 | Not null |

* 1. **MODULE DESCRIPTION**

The main module in this project are listed below

* + - Admin
    - View all applicants
    - Project requirement
    - Apply internship
    - Upload source code

**3.3.1 Admin**

The Company Administration Login consists of the upload all the details of their companies for students and conducting interview details, up to student upload their source code and verify admin process is through online mode.

**3.3.2 View all applicants**

In this module admin checks out the applicants that are registered through the portal and apply for internship.

**3.3.3 Project Requirements**

For the selected candidate’s admin precedes their requirement in project through this portal and gives to student portal

**3.3.4 Apply Internship / Check date**

After final their company where the vacancies of internship are avail, they can apply internship and check the date of interview to proceed.

**3.3.5 Upload Source Code**

Source code is uploaded by the student, after uploaded the final process of receiving completion letter is uploaded by the company administrator and views the letter through this portal or candidate mail.

## CHAPTER 4 SYSTEM TESTING

Testing is an integral part of any system development life cycle. Insufficient and untested applications may tend to crash and the result is loss of economic and manpower investment besides user's dissatisfaction and downfall of reputation. Software testing can be looked upon as one among many processes, an organization performs, and that provides the lost opportunity to correct any flaws in the developed system. Software testing includes selecting test data that have more probability of giving errors.

The first step in system testing is to develop a plan that tests all aspects of the system. Completeness, correctness, reliability and maintainability of the software are to be tested for the best quality assurance that the system meets the specification and requirements for its intended use and performance. System testing is the most useful practical process of executing a program with the implicit intention of finding errors that make the program fails. System testing is done in three phases.

* + - * Unit Testing
      * Integration Testing
      * Validation Testing

### UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software the module. Using the detailed design and the process specification testing is done to registration by the user with in the boundary of the Login module. The login form receives the username and password details and validates the value with the database. If valid, the home page is displayed.

### INTEGRATION TESTING

Integration Testing is the process of this activity can be considered as testing the design and hence module interaction. The primary objective of integration testing is to discover errors in the interfaces between the components. Login form and registration form are integrated and tested together. If the user is newly registered, the received details will be stored in the registration table. While logging in, the application will check for valid user name and password in the registration table and if valid the user is prompted for submitting complaints.

### VALIDATION TESTING

Validation are independent procedures that are used together for checking that a product, service, or system meets [requirements](https://en.wikipedia.org/wiki/Requirement) and [specifications](https://en.wikipedia.org/wiki/Specification_(technical_standard)) and that it fulfills its in purpose the actual result from the expected result for the complaint process. Select the complaint category of the complaint by user. The input given to various forms fields are validated effectively. Each module is tested independently. It is tested that the complaint module fields receive the correct input for the necessary details such as complaint category, complaint id, reference name, complaint description, email for further process.

## CHAPTER 5

* 1. **CONCLUSION**

This project provides a very good communication environment for the instructor and the students who wish to register for an internship class. For the instructor, intern offers a very good environment to monitor a student's progress and avoid unnecessary delays. For the students, Intern provides a good environment that all the students can look at the shared information about companies. Students can read the internship guidelines online and understand all that it takes to successfully complete an internship. The system is very easy to use and any additional requirements can be easily added without affecting, the current information. This is what makes it so convenient to use.

* + - Very user-friendly.
    - Easy accessibility to view the project status.
    - Improves data maintenance.

## APPENDICES APPENDIX I SAMPLE CODE

package com.example.demo.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.demo.dao.ApiDao;

import com.example.demo.response.GetCitizenResponse;

import com.example.demo.response.GetComplaintResponse;

import com.example.demo.service.ApiService;

@RestController

@RequestMapping(value = { "/api" })

public class ApiController {

@Autowired

ApiService service;

@Autowired

ApiDao dao;

@GetMapping("/login/{username}/{password}")

public String login(@PathVariable String username,@PathVariable String password) {

return service.login(username,password);

}

@GetMapping("/add\_student/{name}/{qualification}/{mobile}/{front}/{back}/{ps}/{ds}/{designation}/{username}/{password}")

public String add\_student(@PathVariable String name,@PathVariable String qualification,

@PathVariable String mobile,

@PathVariable String front,

@PathVariable String back,

@PathVariable String ps,

@PathVariable String ds,

@PathVariable String designation,

@PathVariable String username,

@PathVariable String password) {

dao.add\_student(name,qualification,mobile,front,back,ps,ds,designation,username,password);

return "Student Saved Sucessfully";

}

@GetMapping("/add\_requirement/{title}/{desc}/{fe}/{be}/{modules}")

public String add\_requirement(@PathVariable String title,@PathVariable String desc,

@PathVariable String fe,

@PathVariable String be,

@PathVariable String modules) {

dao.add\_requirement(title,desc,fe,be,modules);

return "Project Saved Sucessfully";

}

@GetMapping("/add\_intern/{pid}/{sid}")

public String add\_intern(@PathVariable Integer pid,@PathVariable Integer sid) {

dao.add\_intern(pid,sid);

return "Intern Saved Sucessfully";

}

@GetMapping("/get\_student")

public List<Object[]> get\_student() {

return dao.get\_student();

}

@GetMapping("/get\_project")

public List<Object[]> get\_project() {

return dao.get\_project();

}

@GetMapping("/student\_getproject/{id}")

public List<Object[]> get\_project(@PathVariable Integer id) {

return dao.student\_getproject(id);

}

@GetMapping("/admin\_getproject")

public List<Object[]> admin\_getproject() {

return dao.admin\_getproject();

}

@GetMapping("/view\_applicant")

public List<Object[]> view\_applicant() {

return dao.view\_applicant();

}

@GetMapping("/upload\_code/{pid}/{sid}/{code}")

public String upload\_code(@PathVariable Integer pid,@PathVariable Integer sid,@PathVariable String code) {

dao.upload\_code(pid,sid,code);

return "Code uploaded Sucessfully";

}

}

package com.example.demo.dao;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import javax.transaction.Transactional;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.query.NativeQuery;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

@Repository

@Transactional

public class ApiDao {

@Autowired

SessionFactory sf;

public String login(String username, String password) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select \* from admin where username='"+username+"' and password='"+password+"'";;

NativeQuery nq = session.createNativeQuery(sql);

if (nq.list().size() != 0) {

return "admin";

} else {

String sql1 = "select \* from student where username='"+username+"' and password='"+password+"'";;

NativeQuery nq1 = session.createNativeQuery(sql1);

if (nq1.list().size() != 0) {

List<Object[]> a = nq1.list();

return "id="+a.get(0)[0];

}else {

return "Invalid";

}

}

}

public List<Object[]> get\_student() {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select \* from student";

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

public void add\_student(String name, String qualification, String mobile, String front, String back, String ps,

String ds, String designation,String username,String password) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "INSERT INTO `student` (`id`, `applicantname`, `qualificatino`, `mobile`, `frontend`, `backend`, `ps`, `ds`, `designation`,`username`,`password`) VALUES "

+ "(NULL, '"+name+"', '"+qualification+"', '"+mobile+"', '"+front+"', '"+back+"', '"+ps+"', '"+ds+"', '"+designation+"', '"+username+"', '"+password+"');";

System.out.print("test"+sql);

session.createSQLQuery(sql).executeUpdate();

}

public void add\_requirement(String title, String desc, String fe, String be, String modules) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "INSERT INTO `requirement` (`id`, `title`, `description`, `frontend`, `backend`, `modules`) VALUES "

+ "(NULL, '"+title+"', '"+desc+"', '"+fe+"', '"+be+"', '"+modules+"');";

System.out.print("test"+sql);

session.createSQLQuery(sql).executeUpdate();

}

public void add\_intern(Integer pid, Integer sid) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "INSERT INTO `intern` (`id`, `pid`, `sid`, `code`) VALUES "

+ "(NULL, "+pid+", "+sid+", '');";

System.out.print("test"+sql);

session.createSQLQuery(sql).executeUpdate();

}

public List<Object[]> get\_project() {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select \* from requirement";

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

public List<Object[]> student\_getproject(Integer id) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "SELECT i.id,r.description,r.modules FROM `intern` as i LEFT JOIN student s on(s.id=i.sid) LEFT JOIN requirement r on(r.id=i.pid) WHERE s.id="+id;

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

public List<Object[]> admin\_getproject() {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "SELECT i.id,s.applicantname,r.description,r.modules,i.code FROM `intern` as i LEFT JOIN student s on(s.id=i.sid) LEFT JOIN requirement r on(r.id=i.pid) ";

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

public List<Object[]> view\_applicant() {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select \* from student";

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

public void upload\_code(Integer pid, Integer sid, String code) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "UPDATE `intern` SET `code` = '"+code+"' WHERE `intern`.`id` = "+pid;

System.out.print("test"+sql);

session.createSQLQuery(sql).executeUpdate();

}

}

<html>

<head>

<link rel="stylesheet" href="./style.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"

integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

crossorigin="anonymous"></script>

</head>

<body class="bodys">

<div class="tabs">

<input type="radio" name="tabs" id="tabone" checked="checked">

<label for="tabone">Project Requirement</label>

<div class="tab">

<h1>Project Requirement</h1>

<div class="form-group">

<label>Project Title</label>

<input class="form-control" id="1">

</div>

<div class="form-group">

<label>Desciption</label>

<input class="form-control" id="2">

</div>

<div class="form-group">

<label>Front end</label>

<input class="form-control" id="3">

</div>

<div class="form-group">

<label>Back end</label>

<input class="form-control" id="4">

</div>

<div class="form-group">

<label>Modules</label>

<Textarea class="form-control" id="5" ></Textarea>

</div>

<button id="requirement\_register" type="submit" class="btn btn-primary">Add Project Requirement</button>

</div>

<input type="radio" name="tabs" id="tabtwo">

<label for="tabtwo">View Applicants</label>

<div class="tab">

<h1>View Applicants</h1>

<table class="table table-bordered">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Applicant Name</th>

<th scope="col">Qualification</th>

<th scope="col">Mobile Number</th>

<th scope="col">Front End Skills</th>

<th scope="col">Bank end skills</th>

<th scope="col">Programming languages</th>

<th scope="col">Database Skills</th>

<th scope="col">Designation</th>

</tr>

</thead>

<tbody id="applicant\_body">

</tbody>

</table>

</div>

<input type="radio" name="tabs" id="tabthree">

<label for="tabthree">Check Code</label>

<div class="tab">

<h1>Code check</h1>

<table class="table table-bordered">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Applicant Name</th>

<th scope="col">Project Name</th>

<th scope="col">View Code</th>

<th scope="col">Date & Time</th>

</tr>

</thead>

<tbody id="checkcode">

</tbody>

</table>

<div class="modal fade" id="exampleModalCenter" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalCenterTitle">Code upload</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<textarea id="code" style="width: 100%;height: 400px;"></textarea>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</div>

<input onclick="logout()" type="radio" name="tabs" id="tabfive">

<label onclick="logout()" for="tabfive">Logout</label>

<div class="tab">

<h1>Logout</h1>

</div>

</div>

</body>

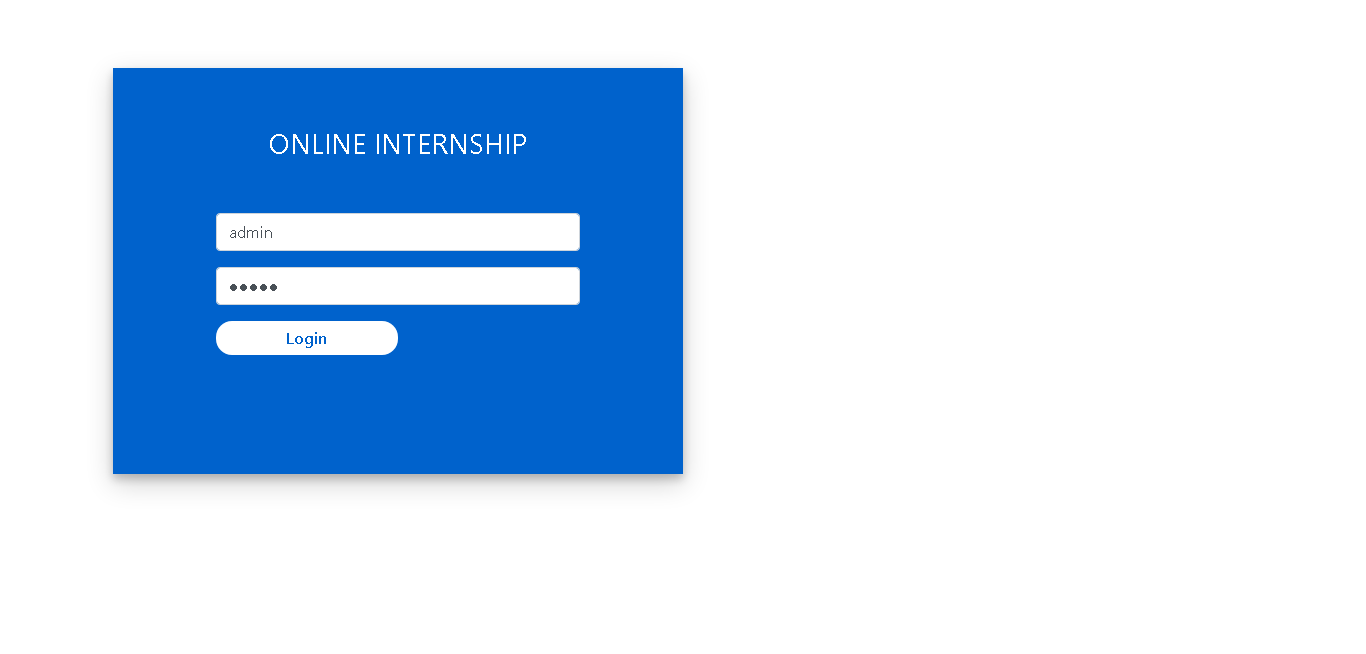
<script src="./admin-home.js">

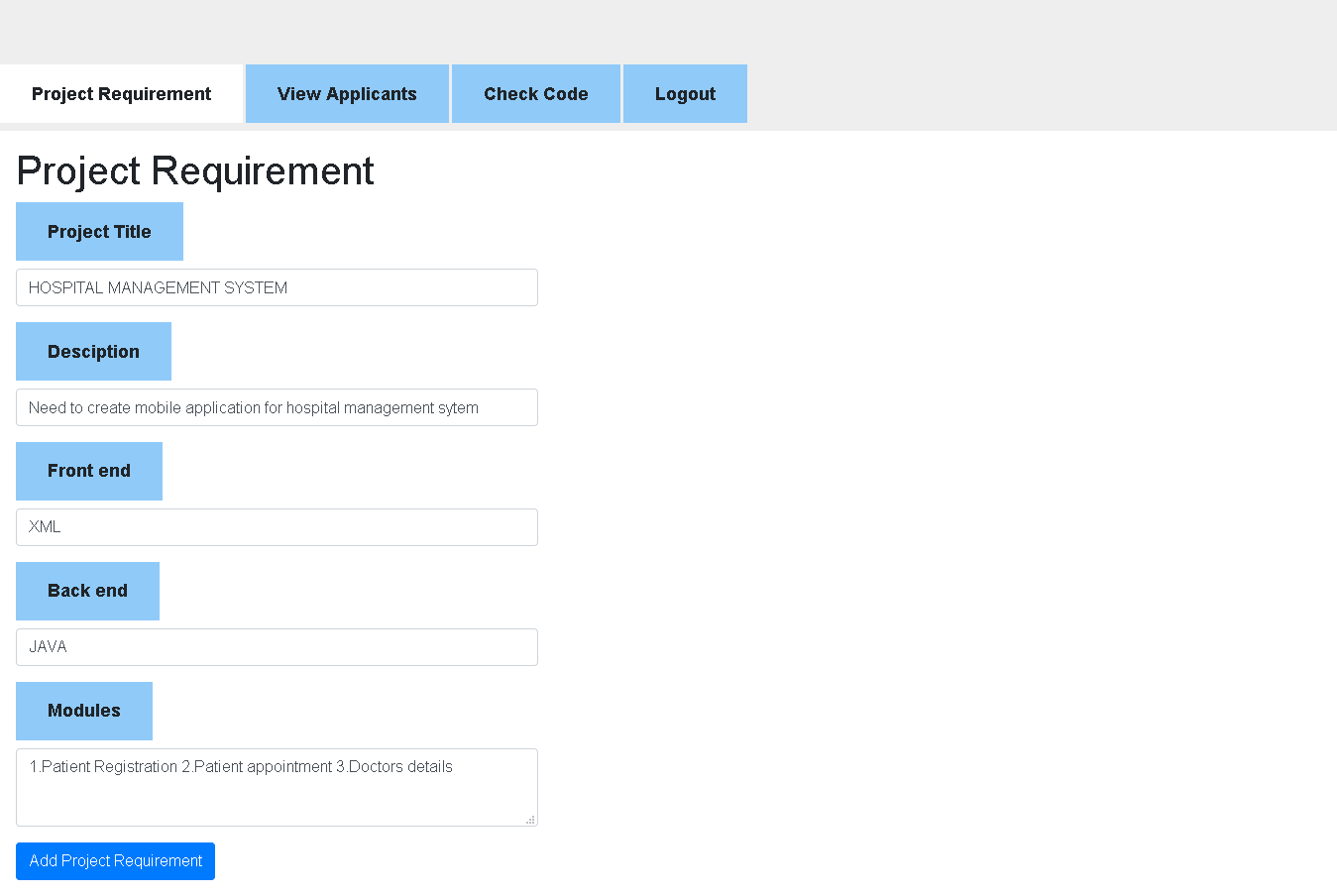
</script>

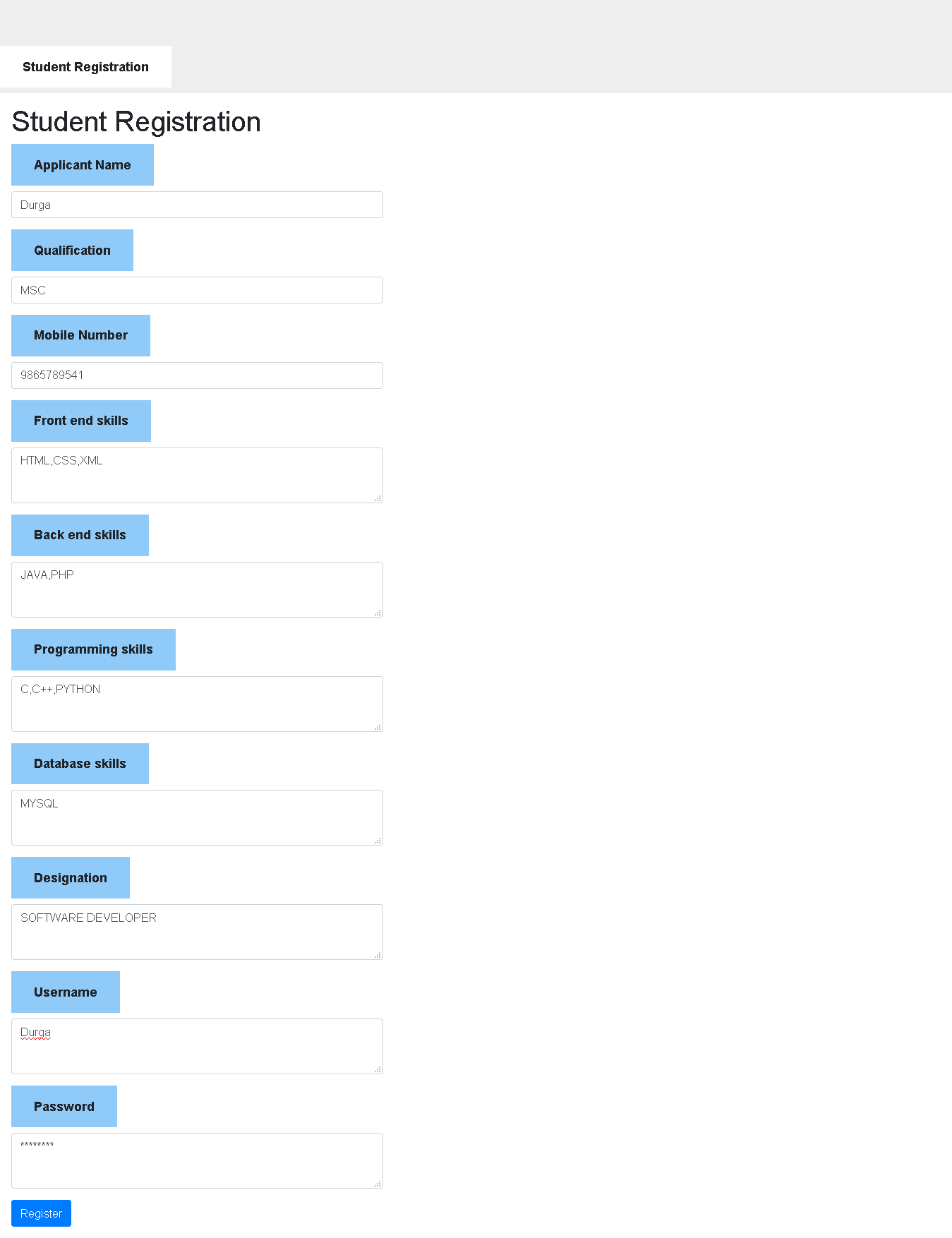
</html>

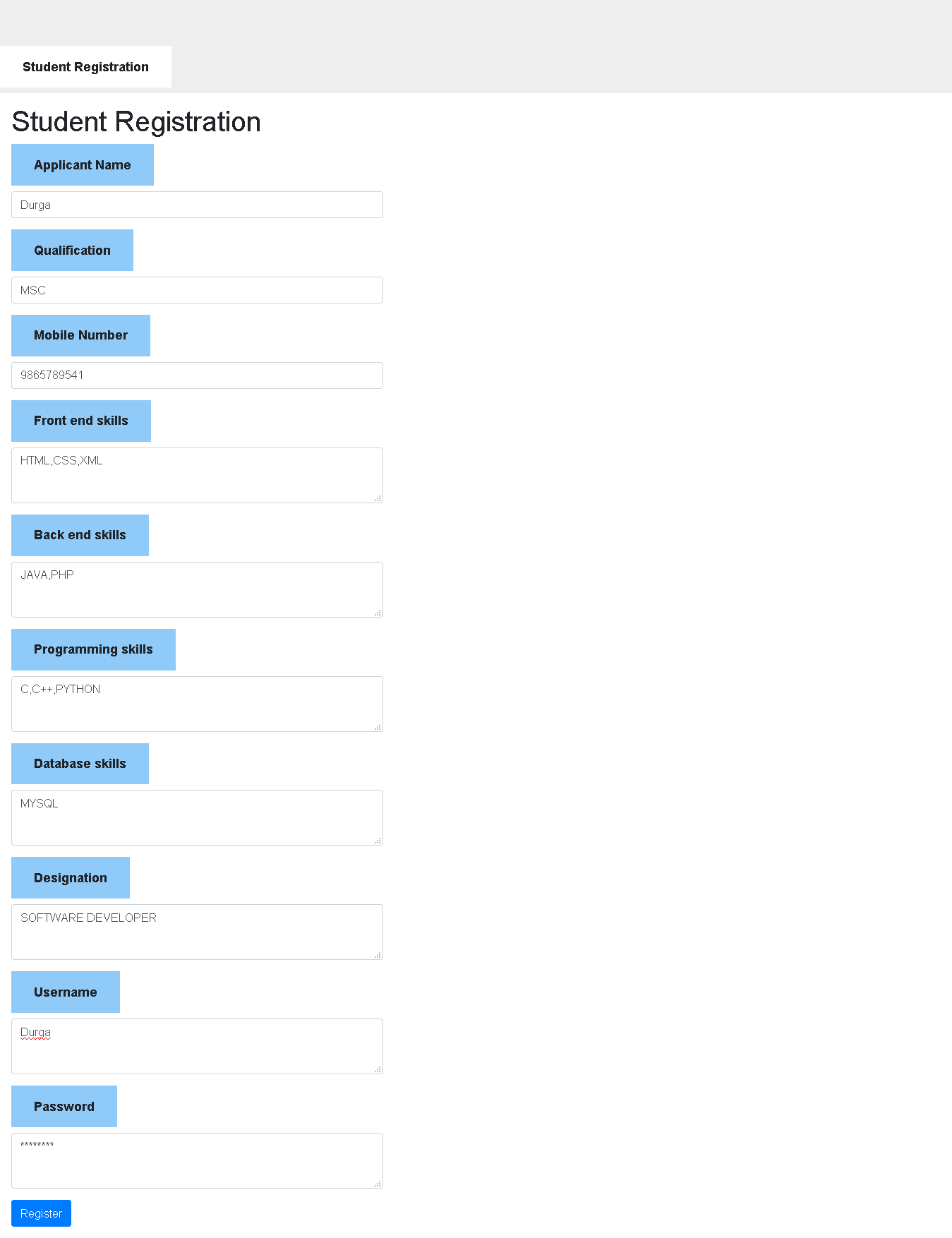
## APPENDIX II

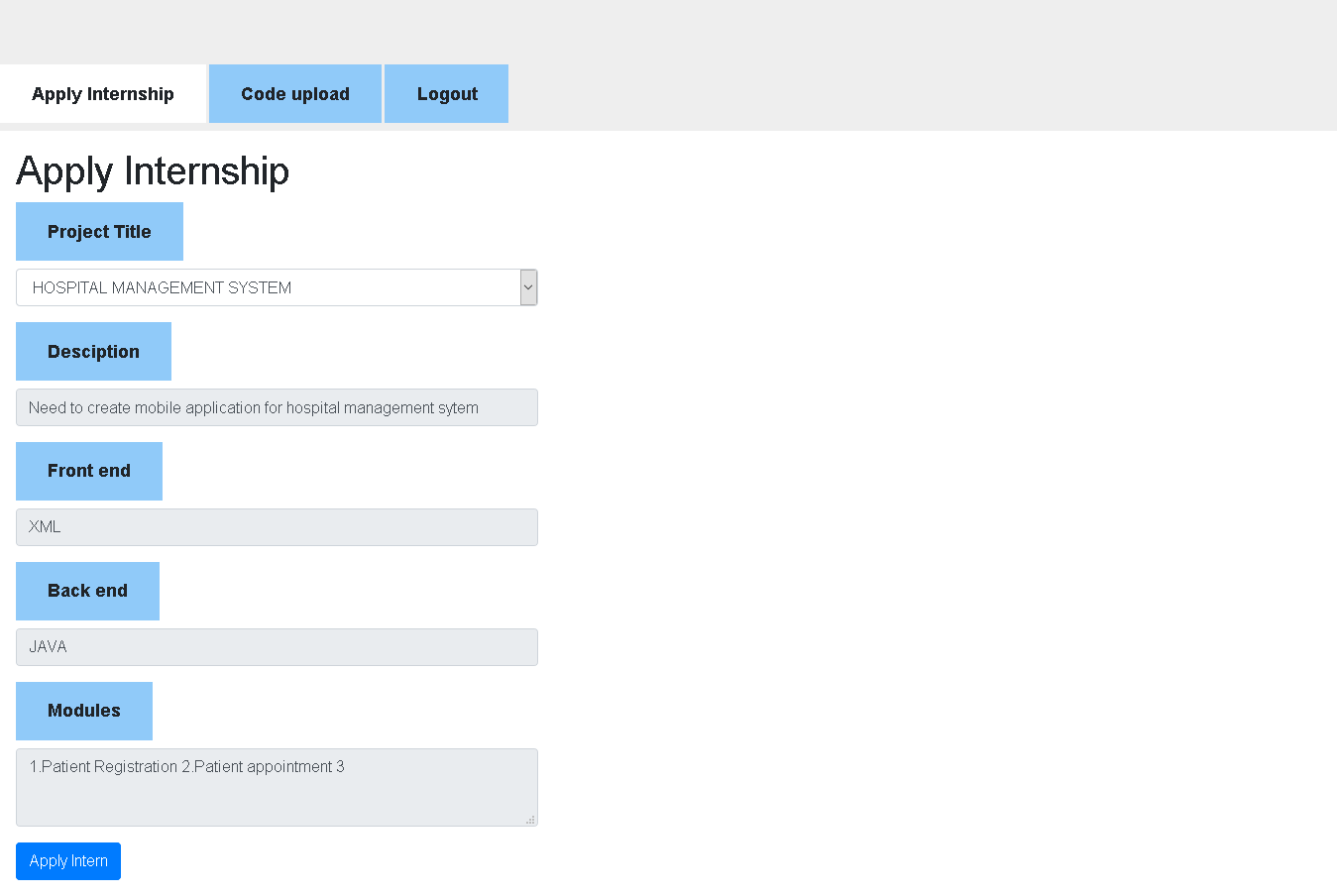
**SAMPLE SCREENSHOTS**

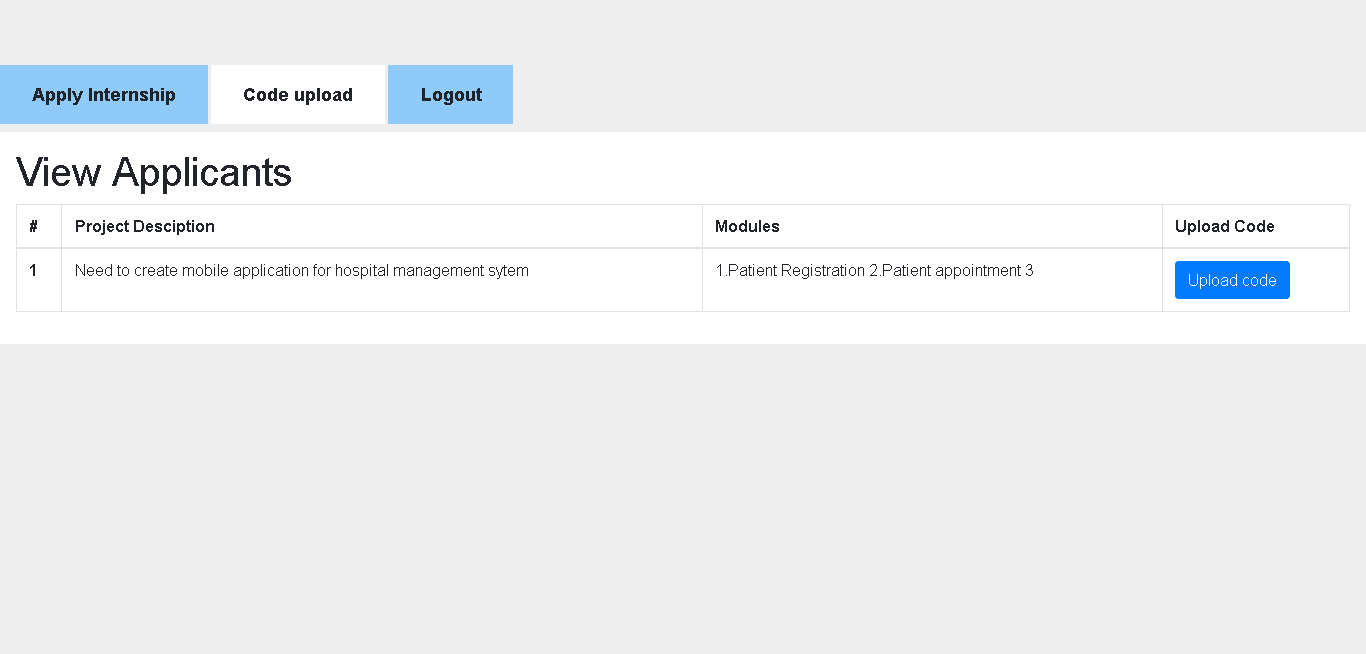


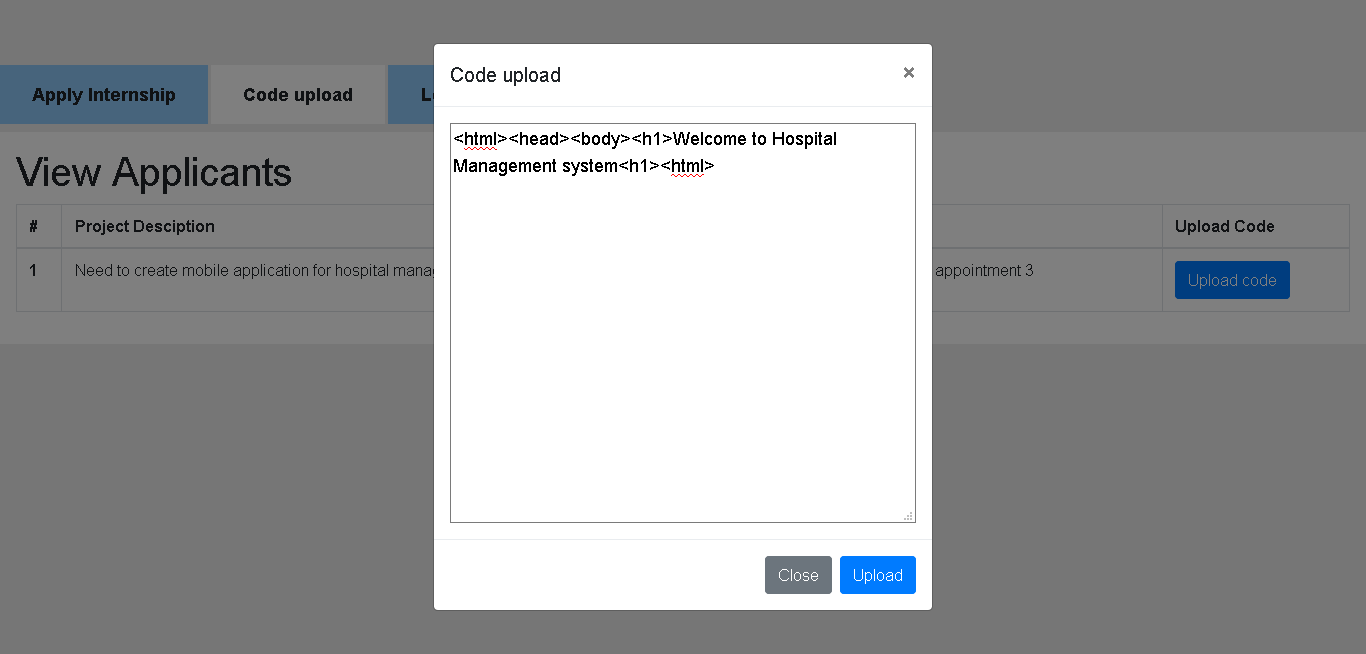


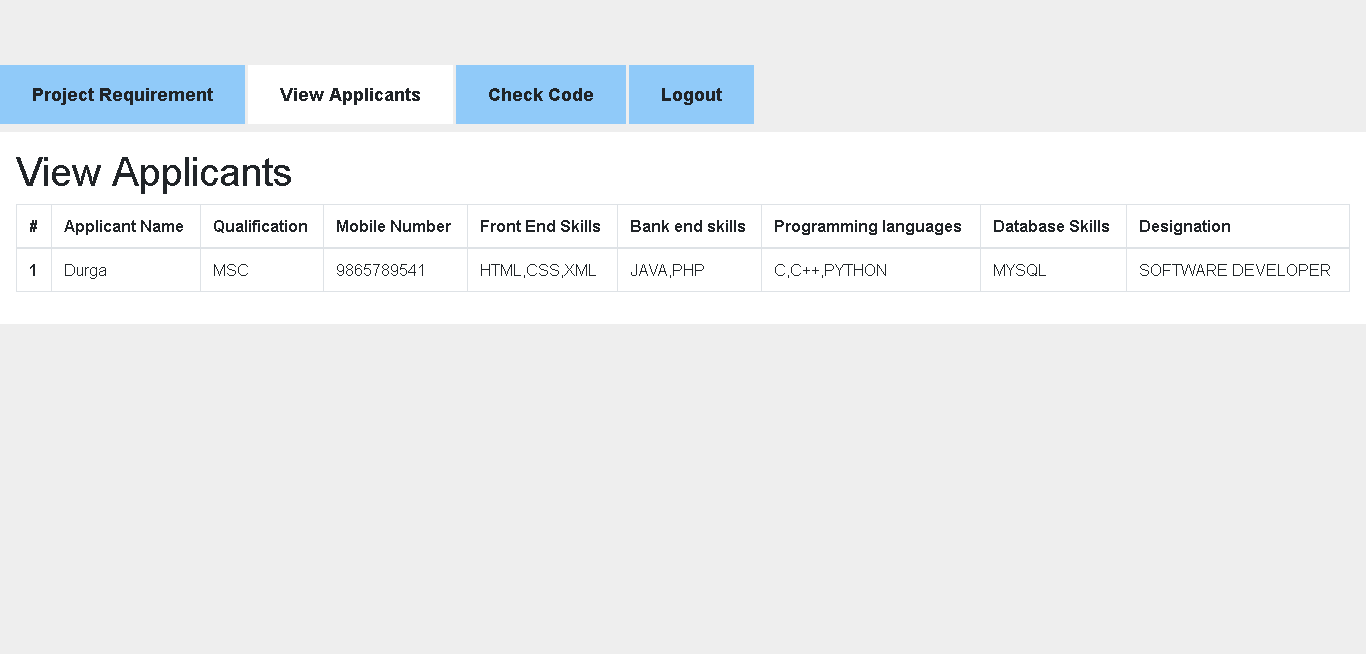


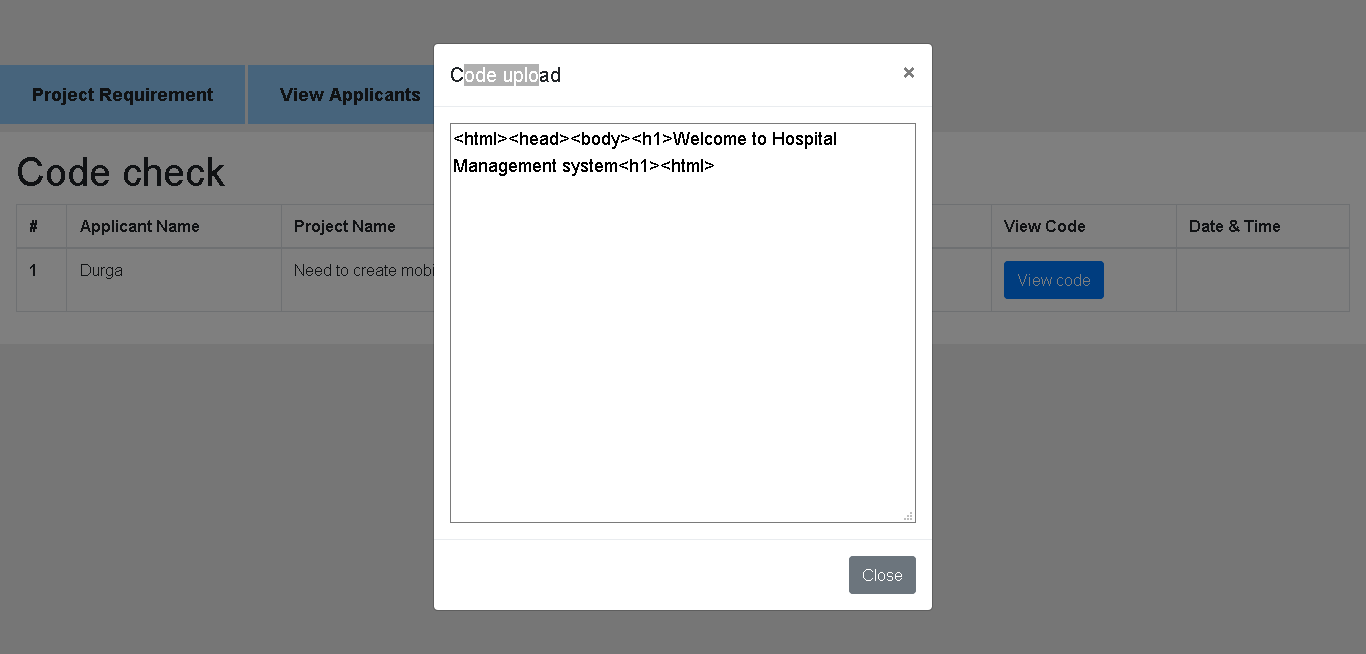












**REFERENCES**

**BOOK REFERENCES**

1. Jeff Forcier, Paul Bissex, Wesley J. Chun, (2018) “Python Web development with Django (Developer’s Library)”, Wesley Professional.
2. Roger S. Pressman, (2017) “Software Engineering”, Tata McGraw-Hill – Europe.

## WEB REFERENCES

1. https:// [www.javaprogramming.net](http://www.javaprogramming.net)
2. https:// [www.w3schools.com](http://www.w3schools.com/)
3. https:// [www.tutorialspoint.com](http://www.tutorialspoint.com/)