**STUDENT PROJECT TRACKING THROUGH WEB BASED APPLICATION**

## CHAPTER 1

**INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

As the college in offline accessibility software, **“STUDENT PROJECT TRACKING THROUGH WEB BASED APPLICATION”** has helped staff to track and view student accessibility policies and practices, and leverage the best in accessibility in marks by this web-based application. In today’s college days with tight deadlines and overloaded work of staff, college organizations often need more than just a web product. The staff can easily find out the project status of each student with the help of our system.

With that in mind, staffs are now pleased to enter the Students project details through the offline software in one hand to enter and track their details. In same way it leads to very easy empowerment of software accessing one by the single staff to track the details very clearly, for this problem we need to track the students efficiently in such a way we have designed the Web based application to track it. The information about student project and their details can be generated as a report through the report option. To know the overall details the maintained information are extracted as report for any use.

The system has introduced for keeping the students project details in tracking manner, now the project has modules such as student’s entry, staff allocation to students as guide, project status tracking of students, at currently student has got how many marks, how many days absent to the review, title submission, guides reports are all view by online and allocation of project also done by online, with corrections, modifications etc.

## CHAPTER 2

**SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM**

Existing system of system of project management is manual. Project coordinator or guide gives task for student manually. Student complete the work which is given by coordinator or guide and submits manually, in this system all work is done by manually so it can take more time to complete project related work. Project coordinator or guide requires remembering in mind when student completed the work so it is difficult for Project coordinator or guide which student completed the task and when. In the existing system does not help users to get right information at right time and user cannot manage project development easily to achieve the main goal. Limitations of existing system

## Drawbacks of Existing System

The following are the drawbacks of the existing system

* It is time consuming
* Right information is not retrieved at right time.
* Any updates to the data by team members or the Project coordinator or guide cannot see immediately by the rest of the team.
* All work is done manually.

## PROPOSED SYSTEM

In this proposed system we can implement a system which can manage project cognate all work consummated by utilized and Project coordinator or guide. Coordinator updates project cognate information, view work done by a student at which time and view progress chart of work done by student, progress chart is developed utilizing Student retrieved the given work information updates and consummates this work at given time and submits into the project management system

## 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

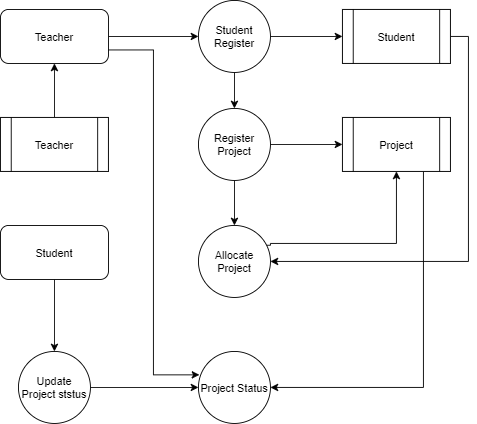
The orders from public are submitted through email, letter or form to the corresponding authority. The registered orders are forwarded to corresponding authority and report about the orders is submitted. The status of orders addressed is not recorded and maintained properly. The maps are not implemented. 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**

### 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 |
| Name | Varchar | 15 | Not null |
| Rollno | Varchar | 15 | Not null |
| Department | Varchar | 15 | Not null |
| Classname | Varchar | 15 | Not null |
| Username | Varchar | 15 | Not null |
| Password | Varchar | 15 | Not null |
| Guide | Varchar | 15 | Not null |
| Mobile | Varchar | 15 | Not null |

**Table 3.3 project**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD TYPE** | **SIZE** | **CONSTRAINT** |
| Id | Int | 10 | Primary key |
| Title | Varchar | 30 | Not null |
| Description | Varchar | 30 | Not null |
| Modules | Varchar | 100 | Not null |
| Technology | Varchar | 100 | Not null |

**Table 3.4 allocate**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **FIELD TYPE** | **SIZE** | **CONSTRAINT** |
| Id | Int | 10 | Primary key |
| Student\_id | Int | 10 | Foreign key |
| project\_id | Int | 10 | Foreign key |
| status | Varchar | 30 | Not null |

* 1. **MODULE DESCRIPTION**

The main module in these projects are listed below

* + - Admin Login
    - Student Registration
    - Create project details & requirement
    - Project allocation
    - Student details
    - Pending & complete project details

**3.3.1 Admin Login**

This module admin can able login into the application and perform an action to managing the student project tracking. Admin can have all the access to do the bellowed modules activities.

**3.3.2 Student Registration Form**

This module admin collects the all the information from the student and stored into the student table. Student table have all the requested field to map the application. Once the student has a mapped before should create the registration form.

* + 1. **Create Project Details and Requirement**

First admin should register the project details and requirement. After then they can able to check the requirement, based on the requirement mean to allocate the student to particular project.

**3.3.4 Project Allocation**

Admin compare the technology wise student to allocate the student to the particular project. Once all the students are allocating into the particular project their list will be showing into the project status window.

* + 1. **Student Details**

Here we can able view the all the student details and allocating project details as well. A cumulative report can we collect here for the user. We can check the student count and student project details.

* + 1. **Pending & completed project status**

This module will show the user for displaying the pending and completed project details. Admin can easily check the current status of the all the project in a single window.

## 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**

Conclusion In the decades since the introduction of QR codes,it has increased rapidly. From the year 1994, many codes have been developed but qr code still remains functional. It is still better than barcodes which are used less than the qr codes and with its pros it tends to get better.

The implementations and features made in the Cargo Management System are as follows,

* + - Very user-friendly.
    - Easy accessibility to view the order 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}/{rollno}/{department}/{classname}/{guide}/{mobile}/{username}/{password}")

public String member\_register(@PathVariable String name,@PathVariable String rollno,

@PathVariable String department,

@PathVariable String classname,

@PathVariable String guide,

@PathVariable String mobile,

@PathVariable String username,

@PathVariable String password) {

dao.studentRegister(name,rollno,department,classname,guide,mobile,username,password);

return "Student Saved Sucessfully";

}

@GetMapping("/add\_project/{title}/{desc}/{modules}/{technology}")

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

@PathVariable String modules,

@PathVariable String technology) {

dao.add\_project(title,desc,modules,technology);

return "Project Created ";

}

@GetMapping("/allocate\_project/{studentid}/{projectid}")

public String allocate\_project(@PathVariable String studentid,@PathVariable String projectid) {

dao.allocate\_project(studentid,projectid);

return "Project Allocated Completed Sucessfully";

}

@GetMapping("/update\_status/{allocate\_id}/{status}")

public String allocate\_project(@PathVariable Integer allocate\_id,@PathVariable String status) {

dao.update\_project(allocate\_id,status);

return "Project Status updated Sucessfully";

}

@GetMapping("/get\_projects")

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

return dao.get\_projects();

}

@GetMapping("/get\_projectsA")

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

return dao.get\_projectsA();

}

@GetMapping("/get\_student")

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

return dao.get\_student();

}

@GetMapping("/get\_student/{id}")

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

return dao.get\_student(id);

}

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 void studentRegister(String name, String rollno, String department, String classname, String guide,

String mobile, String username, String password) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "INSERT INTO `student` (`id`, `name`, `rollno`, `department`, `classname`, `guidename`, `mobile`, `username`, `password`) VALUES "

+ "(NULL, '"+name+"', '"+rollno+"', '"+department+"', '"+classname+"', '"+guide+"', '"+mobile+"', '"+username+"', '"+password+"');";

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

session.createSQLQuery(sql).executeUpdate();

}

public void add\_project(String title, String desc, String modules, String technology) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "INSERT INTO `project` (`id`, `title`, `description`, `modules`, `technology`) VALUES "

+ "(NULL, ' "+title+"', '"+desc+"', '"+modules+"', '"+technology+"');";

session.createSQLQuery(sql).executeUpdate();

}

public void allocate\_project(String studentid, String projectid) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "INSERT INTO `allocate` (`id`, `studentid`, `projectid`, `status`) VALUES "

+ "(NULL, '"+studentid+"', '"+projectid +"', 'Initiated');";

session.createSQLQuery(sql).executeUpdate();

}

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

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select s.id as student,p.id,s.name,s.rollno,p.title,p.description,a.status,p.modules,p.technology from student s left JOIN allocate a on(a.studentid=s.id) left JOIN project p ON(p.id=a.projectid)";

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

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

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select s.id as student,p.id,s.name,s.rollno,p.title,p.description,a.status,p.modules,p.technology,s.guidename,a.id as allocate from student s left JOIN allocate a on(a.studentid=s.id) left JOIN project p ON(p.id=a.projectid) where s.id="+id;

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

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

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "select \* from project";

NativeQuery nq = session.createNativeQuery(sql);

return nq.list();

}

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 update\_project(Integer allocate\_id, String status) {

// TODO Auto-generated method stub

Session session = sf.getCurrentSession();

String sql = "UPDATE `allocate` SET `status` = '"+status+"' WHERE `allocate`.`id` = "+allocate\_id;

session.createSQLQuery(sql).executeUpdate();

}

} package com.example.demo.configuration;

import java.util.Properties;

import javax.sql.DataSource;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.orm.hibernate5.HibernateTransactionManager;

import org.springframework.orm.hibernate5.LocalSessionFactoryBean;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration

@EnableTransactionManagement

public class HibernateConfiguration {

@Value("${db.driver}")

private String DB\_DRIVER;

@Value("${db.password}")

private String DB\_PASSWORD;

@Value("${db.url}")

private String DB\_URL;

@Value("${db.username}")

private String DB\_USERNAME;

@Value("${hibernate.dialect}")

private String HIBERNATE\_DIALECT;

@Value("${hibernate.show\_sql}")

private String HIBERNATE\_SHOW\_SQL;

// @Value("${hibernate.hbm2ddl.auto}")

private String HIBERNATE\_HBM2DDL\_AUTO;

@Value("${entitymanager.packagesToScan}")

private String ENTITYMANAGER\_PACKAGES\_TO\_SCAN;

@Bean

public LocalSessionFactoryBean sessionFactory() {

LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();

sessionFactory.setDataSource(dataSource());

sessionFactory.setPackagesToScan(ENTITYMANAGER\_PACKAGES\_TO\_SCAN);

Properties hibernateProperties = new Properties();

hibernateProperties.put("hibernate.dialect", HIBERNATE\_DIALECT);

hibernateProperties.put("hibernate.show\_sql", HIBERNATE\_SHOW\_SQL);

// hibernateProperties.put("hibernate.hbm2ddl.auto", HIBERNATE\_HBM2DDL\_AUTO);

sessionFactory.setHibernateProperties(hibernateProperties);

return sessionFactory;

}

@Bean

public DataSource dataSource() {

DriverManagerDataSource dataSource = new DriverManagerDataSource();

dataSource.setDriverClassName(DB\_DRIVER);

dataSource.setUrl(DB\_URL);

dataSource.setUsername(DB\_USERNAME);

dataSource.setPassword(DB\_PASSWORD);

return dataSource;

}

@Bean

public HibernateTransactionManager transactionManager() {

HibernateTransactionManager txManager = new HibernateTransactionManager();

txManager.setSessionFactory(sessionFactory().getObject());

return txManager;

}

}

package com.example.demo.configuration;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.CorsRegistry;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@Configuration

@EnableWebMvc

public class WebConfig extends WebMvcConfigurerAdapter {

@Override

public void addCorsMappings(CorsRegistry registry) {

registry.addMapping("/\*\*");

}

}

@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}/{rollno}/{department}/{classname}/{guide}/{mobile}/{username}/{password}")

**public** String member\_register(@PathVariable String name,@PathVariable String rollno,

@PathVariable String department,

@PathVariable String classname,

@PathVariable String guide,

@PathVariable String mobile,

@PathVariable String username,

@PathVariable String password) {

dao.studentRegister(name,rollno,department,classname,guide,mobile,username,password);

**return** "Student Saved Sucessfully";

}

@GetMapping("/add\_project/{title}/{desc}/{modules}/{technology}")

**public** String add\_project(@PathVariable String title,@PathVariable String desc,

@PathVariable String modules,

@PathVariable String technology) {

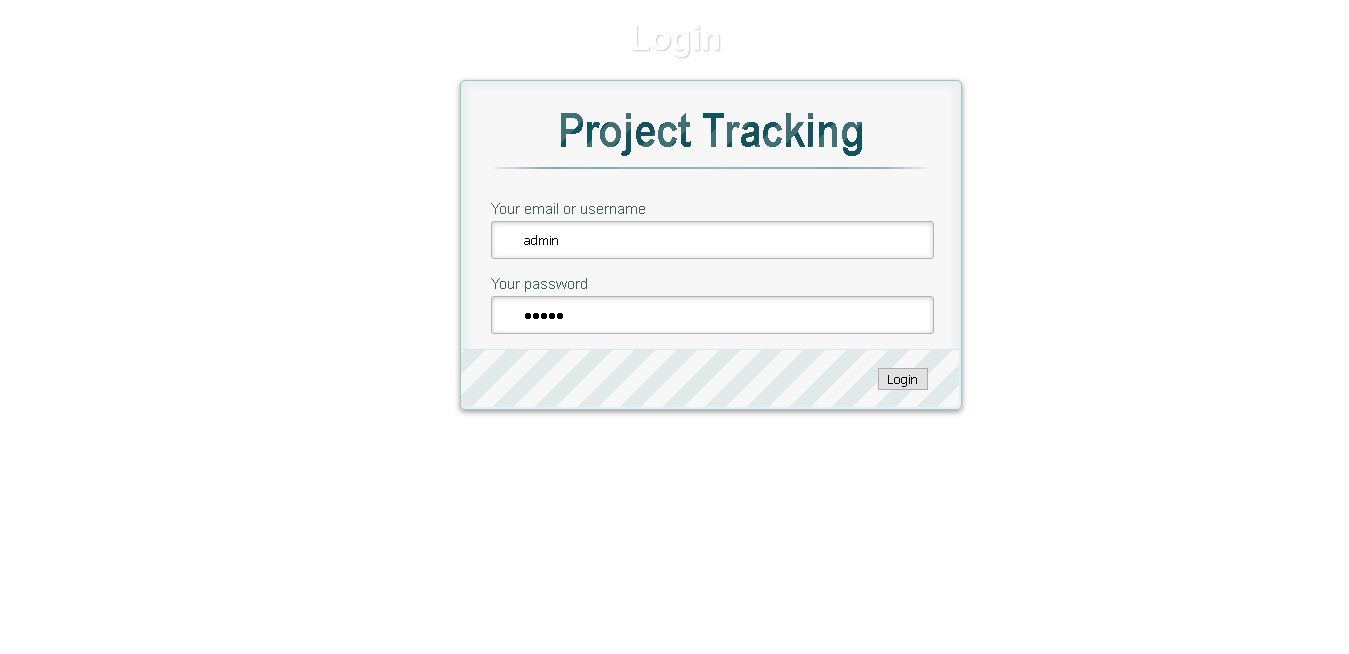
dao.add\_project(title,desc,modules,technology);

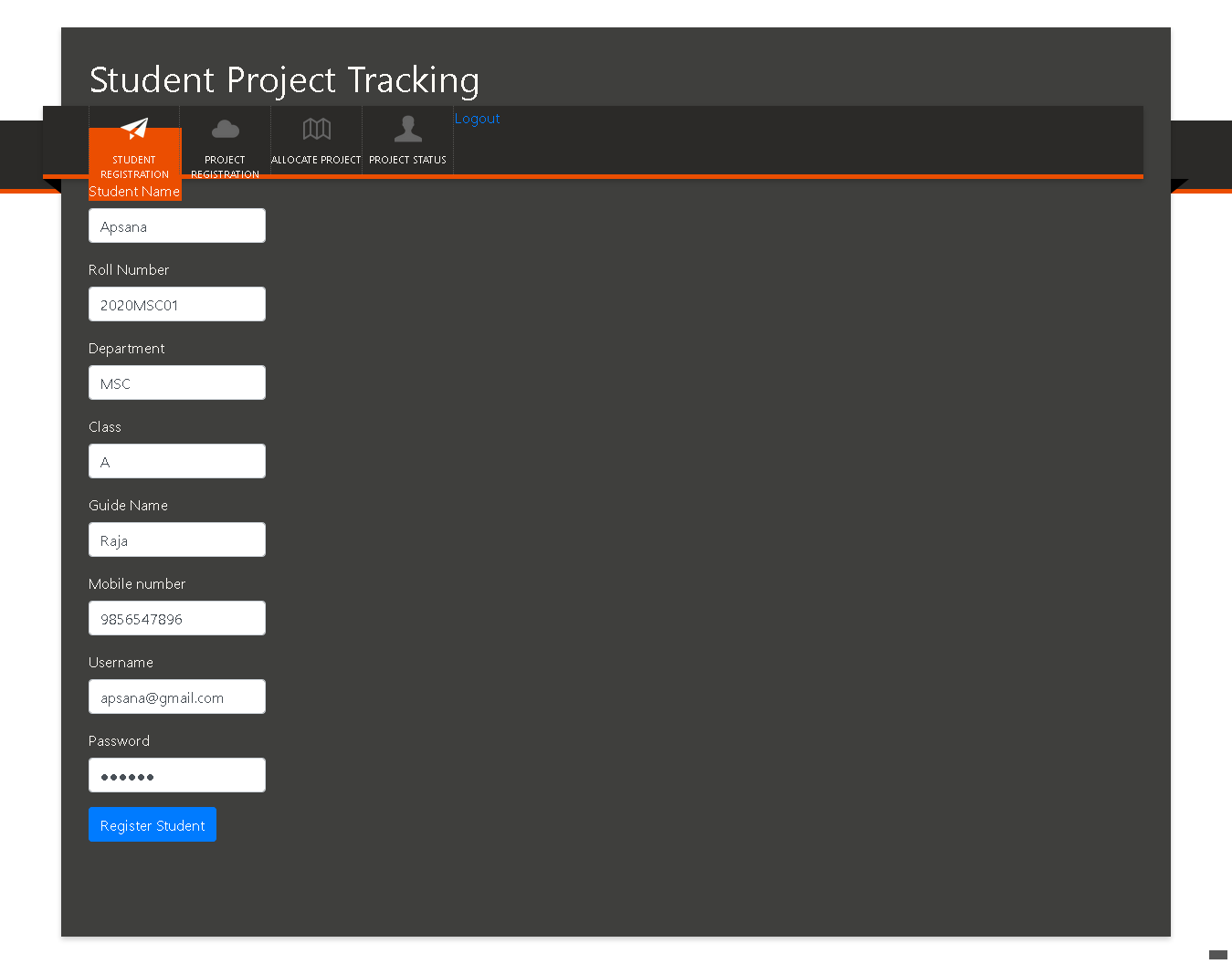
**return** "Project Created ";

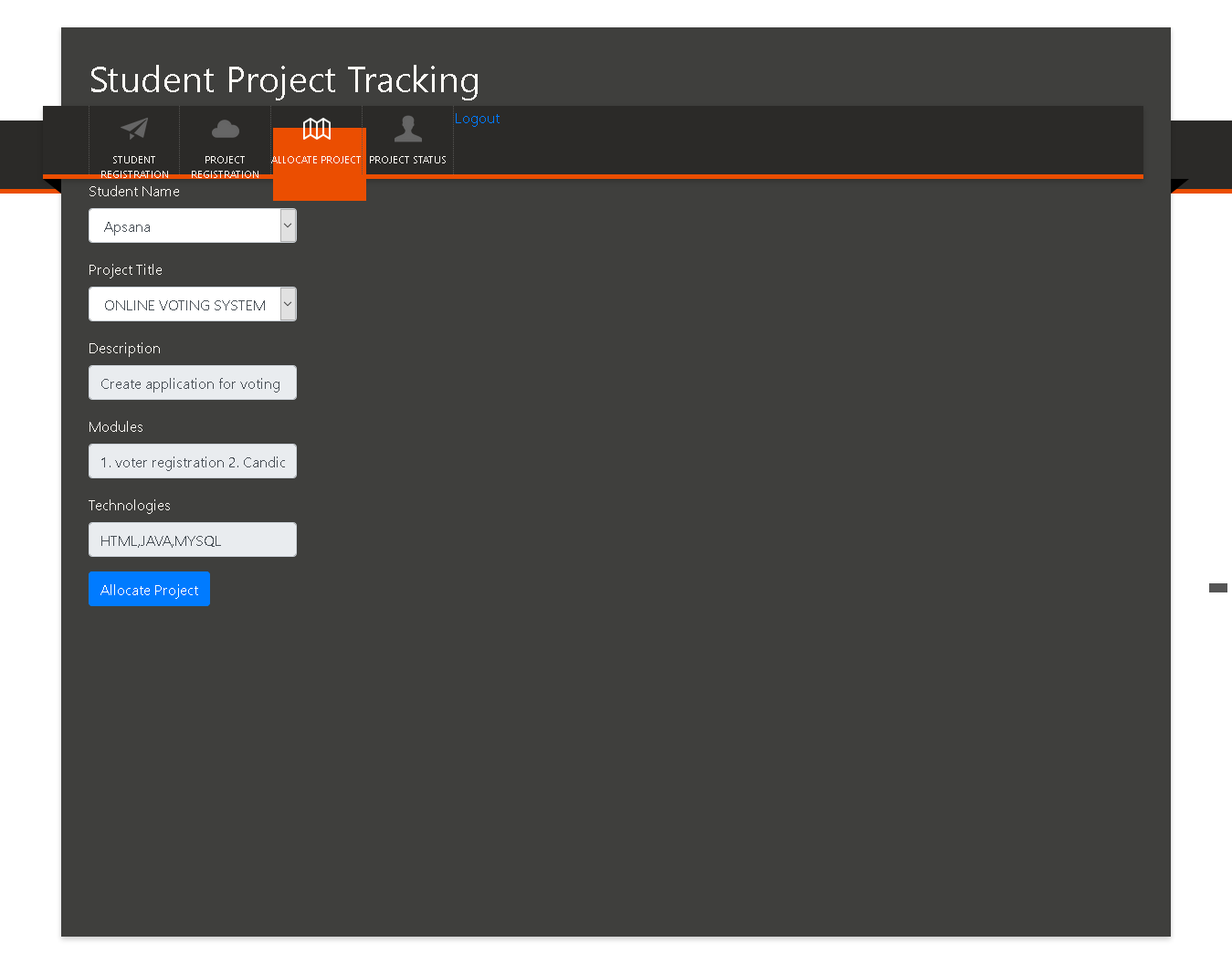
}

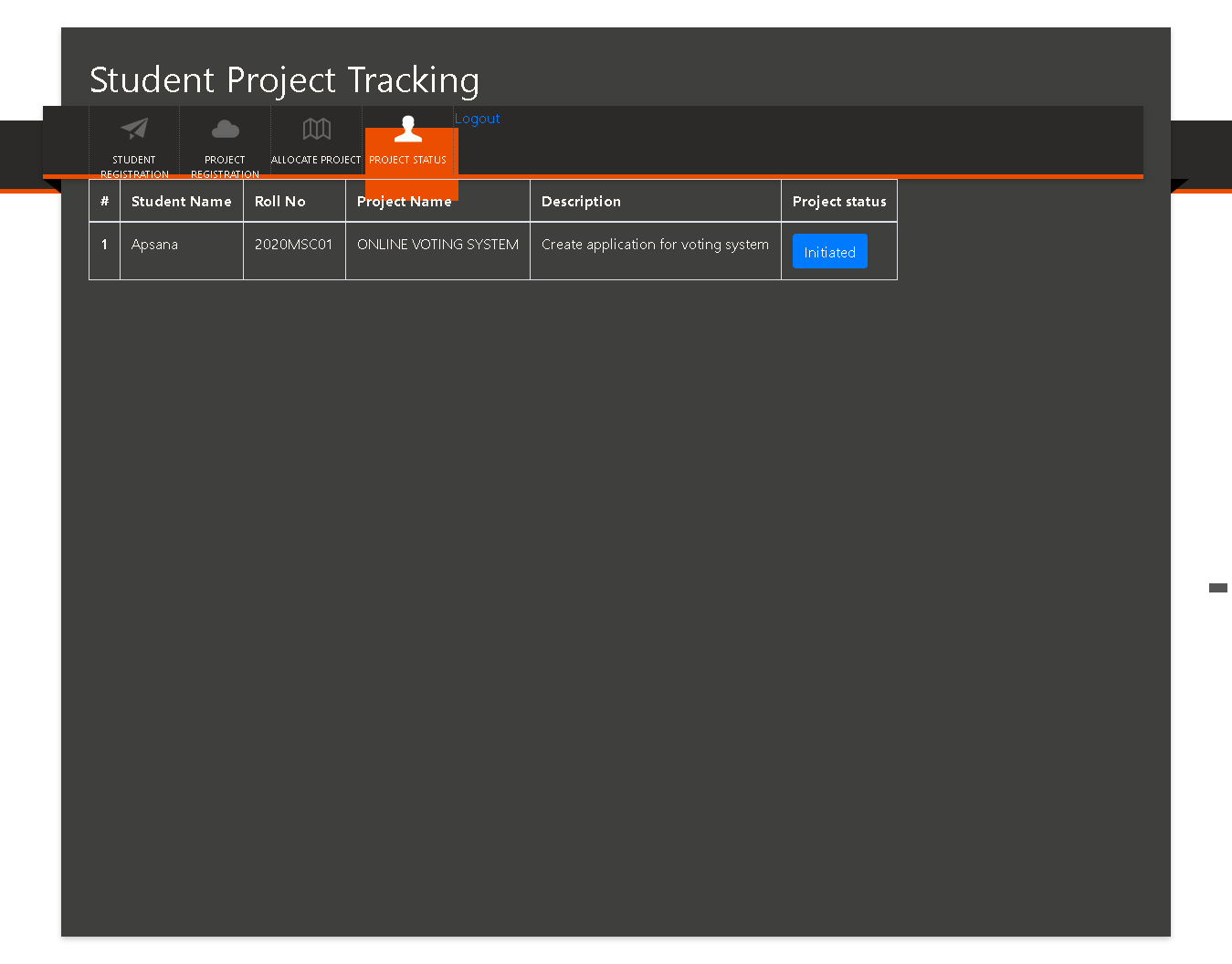
## APPENDIX II

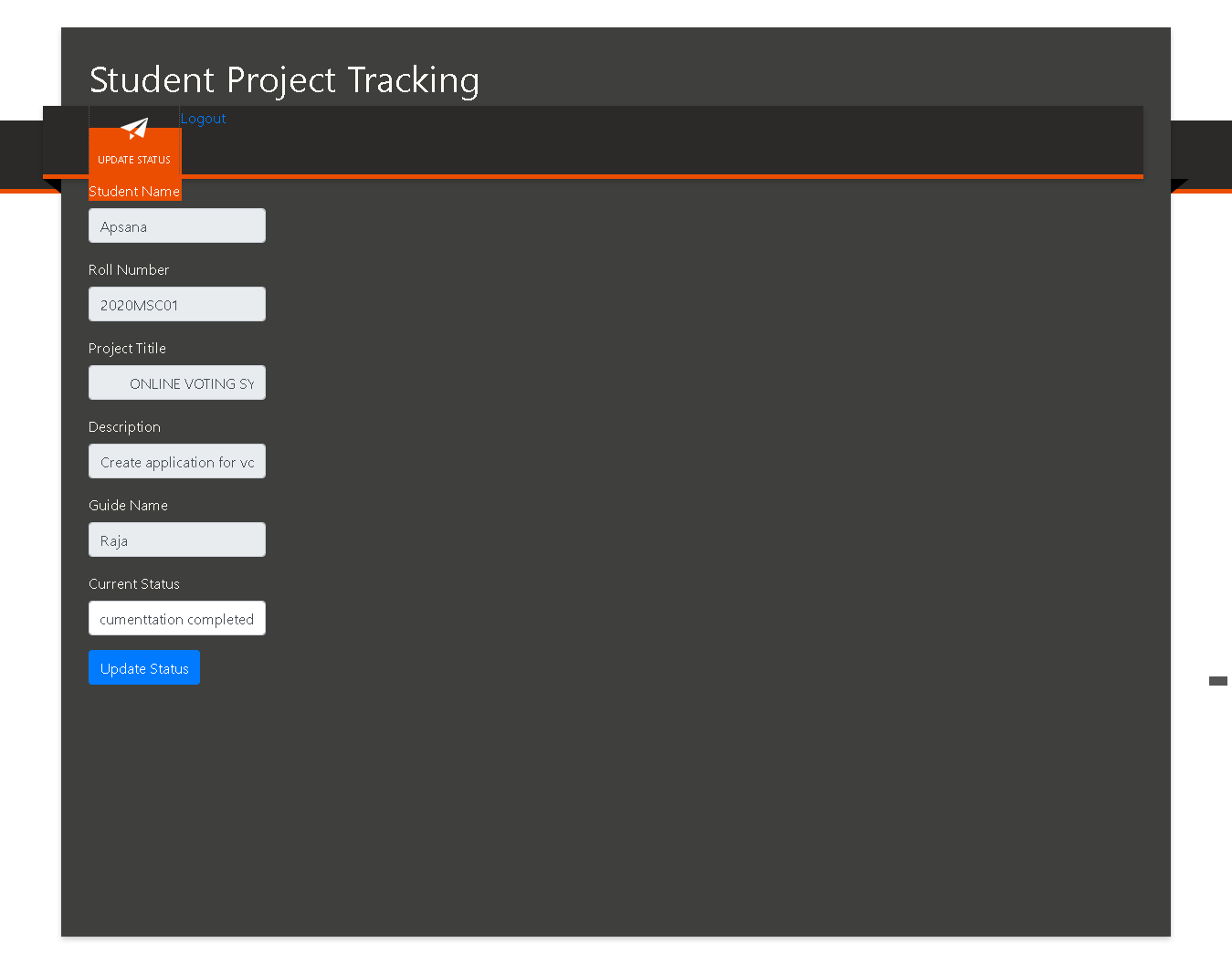
**SAMPLE SCREENSHOTS**

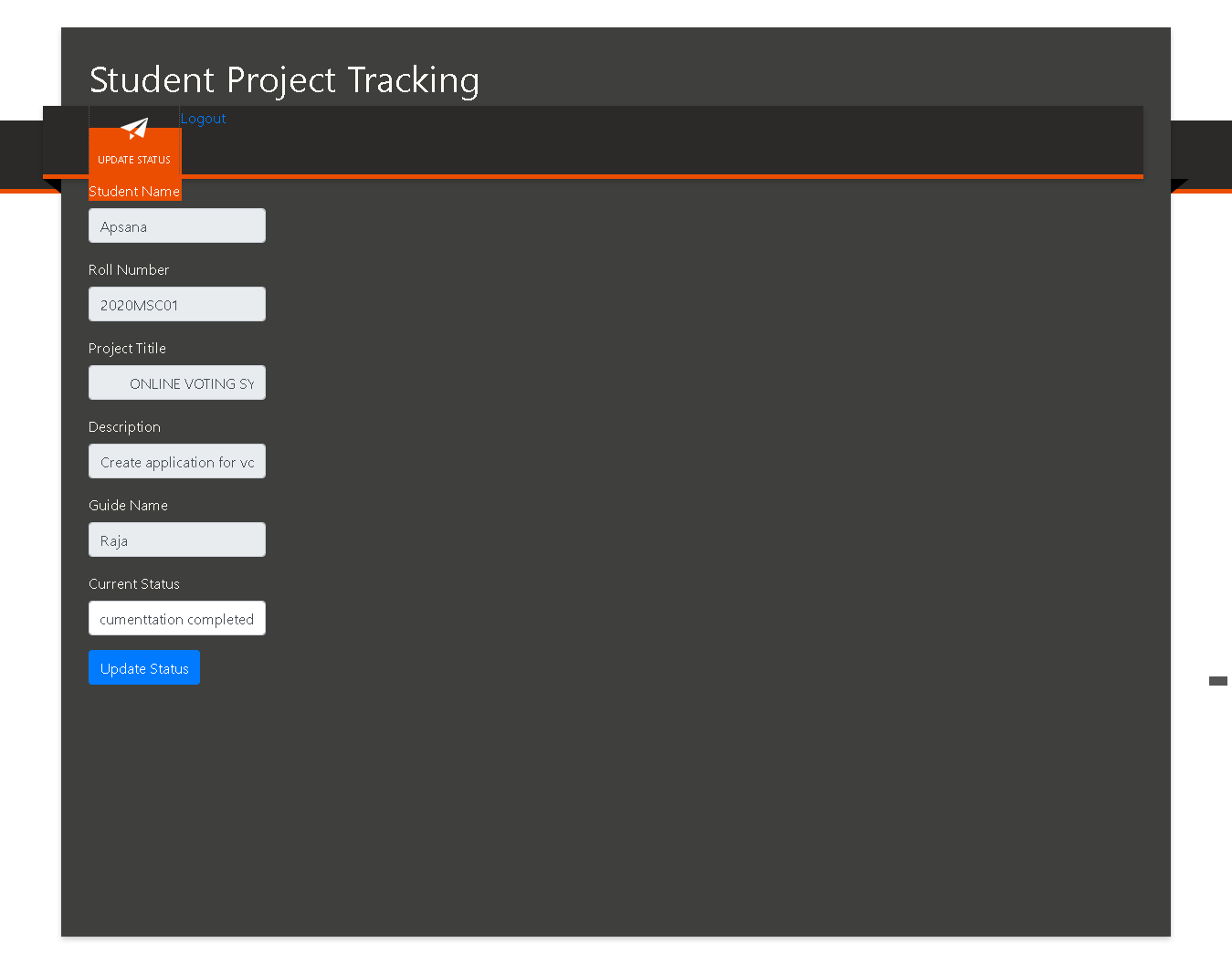












**REFERENCES**

**BOOK REFERENCES**

1. Jeff Forcier, Paul Bissex, Wesley J. Chun, (2018) “Python Web development with Django (Developer’s Library)”, Wesley Professional.
2. Gopala Swamy Ramesh*,* Srinivasan Desikan “Software Testing: Principles and Practice”, Pearson Publications.

## 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/)