PROJECT TRACKING SYSTEM

Submitted in partial fulfillment of requirement for the award of the Degree

Bachelor of Computer Science

In the faculty of Computer Science of Bharathiar University, Coimbatore

Submitted by

N.VEDASRUTI

(Reg.No.2022K0172)

Under the guidance of

Dr A.POORNIMA MCA.,MPhil.,Ph.D

Guest Lecturer, Department of Computer Science

Department of Computer Science

L.R.G GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated To Bharathiar University)

TIRUPUR-4

APRIL-2023

CERTIFICATE

CERTIFICATE

This is to certify that the project work entitled **"PROJECT TRACKING SYSTEM"** Submitted to Bharathiar University in partial fulfilled of the requirement for the award of the Degree of Bachelor of computer science is a record of the original work done by **Ms N.VEDASRUTI (Reg.No.2022K0172)** Under my supervisor and that project work has not formed the basis for the any Degree /Diploma /Association /Fellowship or similar title to any candidate of any university.

Internal Guide	**Head of the Department**
Dr A.POORNIMA MCA.,M.Phil.,Ph.D	**Dr R.PARIMALA M.sc.,M.Phil.,Ph.D**

Viva-voce examination is held on _____L.R.G Government Arts College for Women, Tirupur-641604.

Internal Examiner	**External Examiner**

DECLARATION

DECLARATION

I hereby declare that the project work submitted to the **Department of the Computer Science, L.R.G. Government Arts College for Women, Tirupur**, affiliated to Bharathiar University, Coimbatore in the partial fulfillment of the required for the award of Bachelor of Computer Science is an original work done by me during the sixth semester.

Place: **Signature of the Candidate**

Date: **(N. VEDASRUTI)**

(Reg.No:2022K0172)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

As first and foremost I would like to external my thankfulness to the almighty for blessing the work of my hands, I am grateful to my parents for continued encouragement that they had given to me.

I would like to external my profound gratitude and sincere thanks to **Dr M.R. YEZHILI M.A.,M.Phil.,Ph.D** Principal, L.R.G Government Arts College For Women, for the encouragement rendered to me during this project.

It's my privilege to thank **Dr R. PARIMALA M.Sc.,M.Phil.,Ph.D** Incharge Head of the department of computer science for her valuable guidance and support throughout the project development work.

I express my deep sense of gratitude and sincere thanks to my guide **Dr A. POORNIMA MCA.,M.Phil.,Ph.D** Lecturer, Department of computer science, for providing all sorts of facilities to complete my project work successfully.

I also extend my sincere thanks to all the other faculty member of the department and technical assistance for their co-operation and valuable guidance.

I thank our college library for providing me with many informative books that help me to enrich my knowledge to bring out the project successfully.

CONTENT

CONTENTS

SYNOPSIS

SYNOPSIS

As the college in offline accessibility software, “PROJECT TRACKING SYSTEM” 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.

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.

INTRODUCTION

CHAPTER 1

1.INTRODUCTION

OBJECTIVES

The main objective of the system is easily gathered information from the student to help to track and evaluate the project's progress towards the goals. The given information updates and consummates this work at given time and submit into the project management system.

1.1 ORGANIZATION PROFILE

Retech solution

Currently, we are dealing with three major start-ups built by us. All started with our first start up Retech Projects which provides service and supports for engineering students in their engineering projects and course internships. It is responsible for all the research activities carried by our all other companies. Retech Lasers is created to manufacturing machinery which uses laser technology in their core. We have just started working simultaneously on BLDC motor design and development which is delta by one of our start up called Retech Motors. We are interested to create opportunities for young graduates. Feel free to reach to take part in our mission.

Website

http://www.retechsolutions.in

Industries

Design Services

Company size

11-50 employees

Headquarters

Chennai, Tamil Nadu

Type

Privately Held

1.2 SYSTEM SPECIFICATION

System Requirements Specification also known as Software Requirements Specification, is a document or set of documentation that describes the features and behavior of a software application

WINDOWS OS

Windows is a graphical operating system developed by Microsoft. It allows users to view and store files, run the software, play games, watch videos, and provides a way to connect to the internet. It was released for both home computing and professional works. Microsoft introduced the first version as 1.0

It was released for both home computing and professional functions of Windows on 10 November 1983. Later, it was released on many versions of Windows as well as the current version, Windows 10.

In 1993, the first business-oriented version of Windows was released, which is known as Windows NT 3.1. Then it introduced the next versions, Windows 3.5, 4/0, and Windows 2000. When the XP Windows was released by Microsoft in 2001, the company designed its various versions for a personal and business environment. It was designed based on standard x86 hardware, like Intel and AMD processor. Accordingly, it can run on different brands of hardware, such as HP, Dell, and Sony computers, including home-built PCs.

Microsoft has produced several editions of Windows, starting with Windows XP. These versions have the same core operating system, but some versions included advance features with an additional cost. There are two most common editions of Windows:

- Windows Home
- Windows Professional

Windows Home is basic edition of Windows. It offers all the fundamental functions of Windows, such as browsing the web, connecting to the Internet, playing video games, using office software, watching videos. Furthermore, it is less expensive and comes pre-installed with many new computers.

PYTHON

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMonk found that it was the second-most popular programming language among developers in 2021.

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

"Writing programs is a very creative and rewarding activity," says University of Michigan and Courser instructor Charles R Severance in his book Python for Everybody. "You can write programs for many reasons, ranging from making your living to solving a difficult data analysis problem to having fun to helping someone else solve a problem."

MYSQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Relational database management systems use structured query language (SQL) to store and manage data. The system stores multiple database tables that relate to each other. MS SQL Server, MySQL, or MS Access are examples of relational database management systems. The following are the components of such a system.

A SQL table is the basic element of a relational database. The SQL database table consists of rows and columns. Database engineers create relationships between multiple database tables to optimize data storage space.

SQL statements, or SQL queries, are valid instructions that relational database management systems understand. Software developers build SQL statements by using different SQL language elements. SQL language elements are components such as identifiers, variables, and search conditions that form a correct SQL statement.

1.2.1 HARDWARE CONFIGURATION

- Processor : P 4 700 GHz.
- RAM : 4 GB RAM
- Hard Disk Drive : 180 GB
- Speed : 1 GHz
- Keyboard : Multimedia Keyboard
- Mouse : Optical Mouse
- LED Monitor : 22 inches

1.2.2 SOFTWARE SPECIFICATION

- Operating System : Windows 10
- Front End : PYTHON
- Back End : MYSQL

SYSTEM STUDY

CHAPTER 2

SYSTEM STUDY

2.1 EXISTING SYSTEM

Existing system of project tracking 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.

2.1.1 DRAWBACKS OF 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.

2.2 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 of the student can retrieved the given work information updates and consummates this work at given time and submits into the project management system.

2.2.1 FEATURES OF PROPOSED SYSTEM

- The staff can easily find out the project status of each student with the help of our system.
- To get a Real time information
- Easy to identifies the problem.
- Easy and precise reporting

SYSTEM DESIGN AND DEVELOPMENT

CHAPTER 3

SYSTEM DESIGN AND DEVELOPMENT

3.1 FILE DESIGN

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

A file system is the data structure designed to support the abstraction of the data blocks as an archive and collection of files. This data structure is unique because it is stored on secondary storage (usually the disk), which is a very slow device.

The file system structure is the most basic level of organization in an operating system. Almost all of the ways an operating system interacts with its users, applications, and security model are dependent upon the way it organizes files on storage devices.

File Design Information systems in business are file and database oriented. Data are accumulated into files that are processed or maintained by the system. The systems analyst is responsible for designing files, determining their contents and selecting a method for organizing the data.

The most important purpose of a file system is to manage user data. This includes storing, retrieving and updating data. Some file systems accept data for storage as a stream of bytes which are collected and stored in a manner efficient for the media.

3.2 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:'

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

FEATURES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. 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. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
- will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

3.3 OUTPUT DESIGN

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

External Outputs

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

Internal outputs

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

3.4 DATABASE DESIGN

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

Business needs addressed:

- Determine the basic objects about which the information is stored
- Determine the relationships between these groups of information and the objects
- Effectively manage data and create intelligent information
- Remote database administration or on site administrative support
- Database creation, management, and maintenance
- Information retrieval efficiency, remove data redundancy and ensure data security

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 is 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 accesses 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.

Data Independence

Data in dependence 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.

3.5 SYSTEM DEVELOPMENT

Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software.

Systems development life cycle phases include planning, system analysis, system design, development, implementation, integration and testing, and operations and maintenance.

3.5.1 DESCRIPTION OF MODULES

There are six modules in the project.

- Admin or teacher login
- Student Registration
- Project Registration
- Project Allocation
- Update Status
- Project Status

Admin or teacher 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.

Student registration

This module admin collects 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.

Project registration

Here the admin can register the overall projects. And also add the requirements of project and the required modules.

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.

Update status

This module student can login by using the username and password then the allocated project details are displayed. Student can update their status about the project.

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.

TESTING AND IMPLEMENTATION

CHAPTER 4

TESTING AND IMPLEMENTATION

TESTING METHODOLOGIES

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfill this objective a series of test step unit, integration. Validation and system tests were planned and executed the test steps are:

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.

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

VALIDATION TESTING

Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfills its in purpose the actual result from the expected result for the process. The input given to various forms fields are validated effectively. Each module is tested independently. It is tested that the student registration module fields receive the correct input for the necessary details such as student id, name, Department, class, guided name, mobile number, username and password for further process

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the admin. After validation test has been conducted, one of two possible conditions exists.

OUTPUT TESTING

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

SYSTEM IMPLEMENTATION

When the initial design was done for the system, the client was consulted for the acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about the working of the system. The aim of the system illustration was to identify any malfunction of the system.

After the management of the system was approved the system implemented in the concern, initially the system was run parallel with existing manual system. The system has been tested with live data and has proved to be error free and user friendly.

Implementation is the process of converting a new or revised system design into an operational one when the initial design was done by the system; a demonstration was given to the end user about the working system.

This process is uses to verify and identify any logical working of the system by feeding various combinations of test data. After the approval of the system by both end user and management the system was implemented.

System implementation is made up of many activities. The six major activities are as follows.

CODING

Coding is the process of whereby the physical design specifications created by the analysis team turned into working computer code by the programming team. A design code may be a tool which helps ensure that the aspiration for quality and quantity for customers and their requirements, particularly for large scale projects by the project tracking Design pattern are documented tried and tested solutions for recurring problems in a given context. So basically you have a problem context and the proposed solution for the same.

INSTALLATION

Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, and documentation and work procedures to those consistent with the new system.

DOCUMENTATION

Documentation is descriptive information that describes the use and operation of the system. The user guide is provided to the end user as the student and administrator. The documentation part contains the details as follows,

User requirement and administration has been made online. The college can request their project requirement details through online and also use of documentation, they can view the purpose of each purpose, The admin could verify the authentication of the users, users requirements and need to take delivery process, thus the documentation is made of full view of project thus it gives the guideline to study the project and how to execute also.

USER TRAINING AND SUPPORT

The software is installed at the deployment environment, the developer will give training to the end about the software. The goal of an software is the coordinator updates project cognate information, view work done by a student at which time and view progress of the Student retrieved the given work information updates and consummates this work at given time and submits into the project management system. The following are the instruction which is specified the handling and un-handling events in the application**,**

- The authenticated user of admin and Student only login in the application with authorized username and password.
- Don’t make user waste their time because it processed like systematically.
- It can easily track through online by the user.
- Very user friendliness software

IMPLEMENTATION PROCEDURES

Implementation includes all the activities that take place to convert the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organization requirements. Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

IMPLEMENTATION PROCEDURES

PILOT RUNNING

Processing the current data by only one user at a time called the pilot running process. When one user is accessing the data at one system, the system is sets to be engaged and connected in network. This process is useful only in system where more than one user is restricted.

PARALLEL RUNNING:

Processing the current data by more than one user at a time simultaneously is said to be parallel running process. This same system can be viewed and accessed by more than one user at the time. Hence the implementation method used in the system is a pilot type of implementation.

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

The stage consists of,

- Testing the developed program with sample data.
- Detection's and correction of error.
- Creating whether the system meets user requirements.
- Making necessary changes as desired by the user.
- Training user personnel.

USER TRAINING

User Training is designed to prepare the user for testing &consenting the system.

- User Manual.
- Help Screens.
- Training Demonstration.

USER MANUAL

The summary of important functions about the system and software can be provided as a document to the user.

HELP SCREENS

This features now available in every software package, especially when it is used with a menu. The user selects the “Help” option from the menu. The system accesses the necessary description or information for user reference.

TRAINING DEMONSTRATION:

Another User Training element is a Training Demonstration. Live demonstrations with personal contact are extremely effective for Training Users.

SYSTEM MAINTENANCE

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend for more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits**.**

- Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
- Users are not fully cognizant of the maintenance problem or its high cost.
- Few tools and techniques are available for maintenance.
- A good test plan is lacking.
- Standards, procedures, and guidelines are poorly defined and enforced.
- Programs are often maintained without care for structure and documentation.
- There are minimal standards for maintenance.
- Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

Corrective Maintenance

It means repairing, processing or performance failure or making changes because of previously uncovered problems or false assumptions. Task performed to identify, isolate, and rectify a fault so that the failed equipment, machine, or system can be restored to an operational condition within the tolerances or limits established for in-service operations.

Corrective maintenance can be subdivided into "immediate corrective maintenance" (in which work starts immediately after a failure) and "deferred corrective maintenance" (in which work is delayed in conformance to a given set of maintenance rules).

Perfective Maintenance

It means changes made to a system to add new features or to improve performance. Preventive maintenance is predetermined work performed to a schedule with the aim of the project Coordinator updates project cognate information, view work done by a student at which time and view progress of the Student can retrieved the given work information updates and consummates this work at given time and submits into the project management system.

Time-based or run-based Periodically inspecting, servicing, cleaning, or replacing parts to prevent sudden failure .On-line monitoring of equipment in order to use important/expensive parts to the limit of their serviceable life. Preventive maintenance involves changes made to a system to reduce the chance of future system failure.

An example of preventive maintenance might be to increase the number of records that a system can process far beyond what is currently needed or to generalize how a system sends report information to a printer so that so that the system can adapt to changes in printer technology.

Preventive Maintenance

Changes made to a system to avoid possible future problems Perfective maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. The objective of perfective maintenance is to improve response time, system efficiency, reliability, or maintainability.

During system operation, changes in user activity or data pattern can cause a decline in efficiency, and perfective maintenance might be needed to restore performance. Usually, the perfective maintenance work is initiated by the department, while the corrective and adaptive maintenance work is normally requested by users.

CONCLUSION

CHAPTER 5

CONCLUSION

Student project tracking systems are an important tool for educational institutions, helping students to manage their projects effectively and achieve their academic goals. These systems allow students to track the progress of their projects, collaborate with other team members, and receive feedback and guidance from their instructors.

One of the biggest advantages of student project tracking systems is the increased transparency they offer. By providing students with real-time updates on the status of their projects, these systems help to keep them on track and ensure that they are meeting their deadlines. This also allows instructors to monitor student progress more effectively and provide guidance and support where needed.

Another advantage of student project tracking systems is the increased collaboration they facilitate. By allowing students to communicate and collaborate with other team members in real-time, these systems make it easier for students to work together effectively and share ideas and feedback.

Student project tracking systems also offer benefits for instructors and educational institutions as a whole. By providing a centralized platform for managing student projects, these systems can reduce administrative overhead and make it easier for instructors to manage large groups of students.

Overall, student project tracking systems are an essential tool for educational institutions, helping students to manage their projects effectively and achieve their academic goals. As technology continues to evolve, these systems will likely become even more advanced and provide even more benefits for students and instructors alike.

BIBLIOGRAPHY

BIBLIOGRAPHY

References:

- Rossum, Guido van. "Python Programming Language." Python.org, https://www.python.org/.
- Summerfield, Mark. Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional, 2009.
- Hill, Christian, and Harris, Chad. Learning Scientific Programming with Python. Cambridge University Press, 2016.
- Beazley, David M. Python Essential Reference. Addison-Wesley Professional, 2009.
- Brownlee, Jason. Machine Learning Mastery with Python. Machine Learning Mastery, 2016.

Website:

- Python Software Foundation. "Python Package Index (PyPI)." Python.org, https://pypi.org/.
- Python Software Foundation. "The Python Standard Library." Python.org, https://docs.python.org/3/library/.
- Real Python. "Python Debugging With Pdb." Real Python, https://realpython.com/python-debugging-pdb/.
- Python.org. "Official Python Documentation." Python.org, https://www.python.org/doc/.
- Real Python. "Python Tutorials, Articles and News." Real Python, https://realpython.com/.

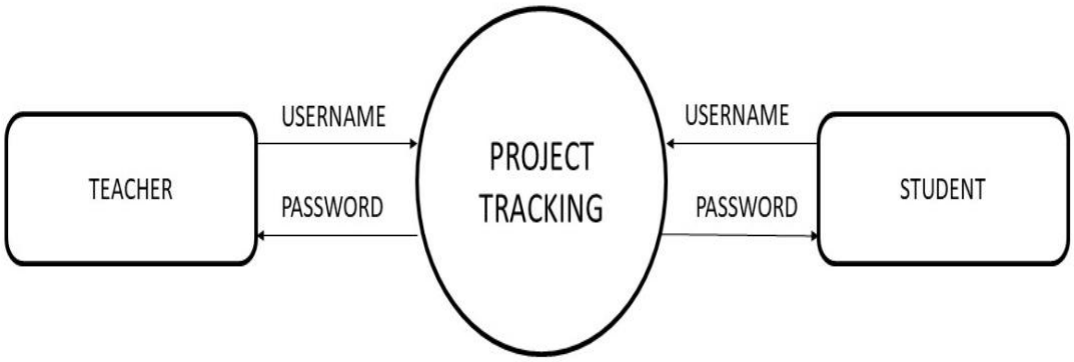
APPENDICES

APPENDICES

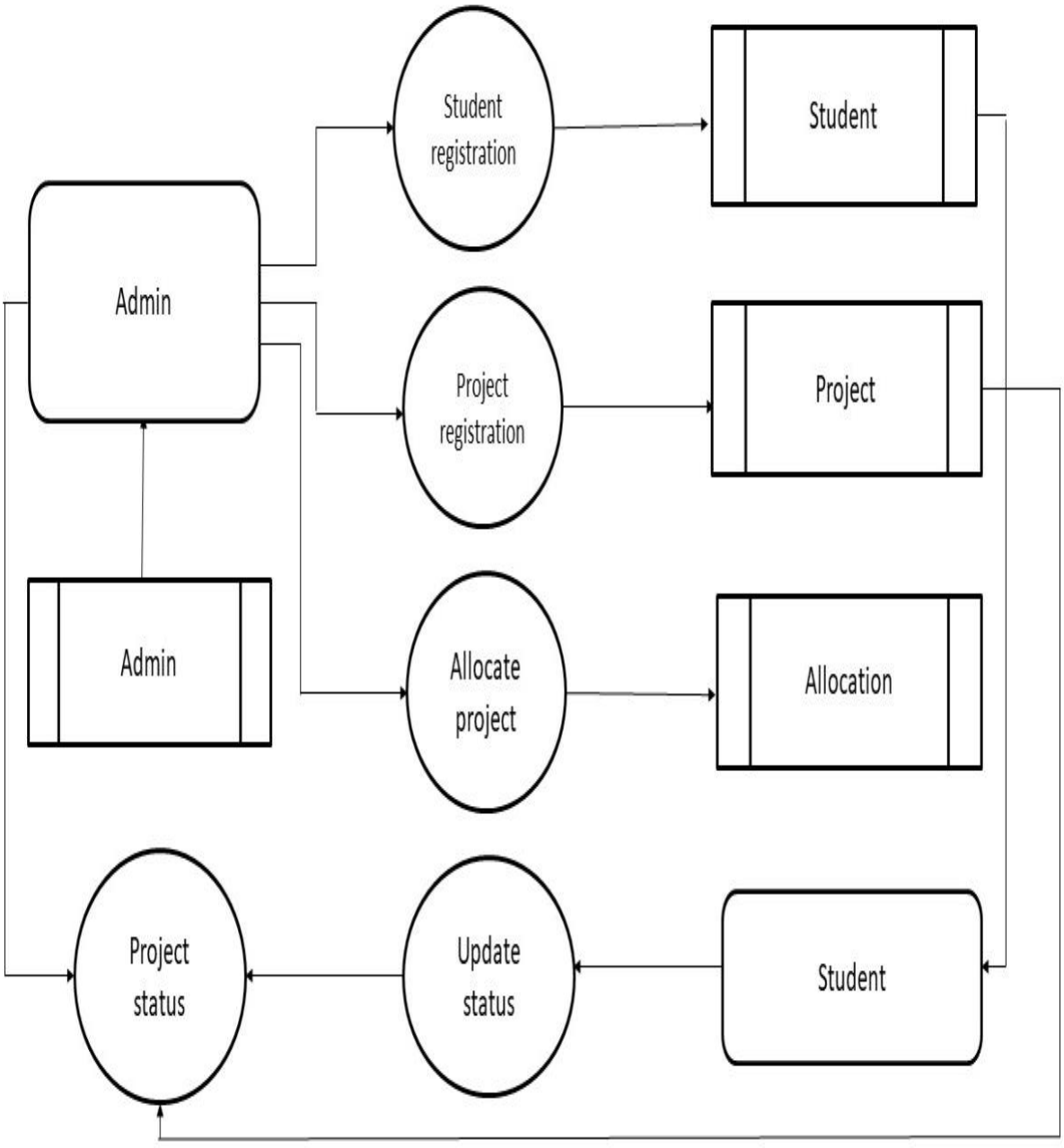
A. DATA FLOW DIAGRAM

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 modeling tools.

LEVEL 0:



LEVEL 1:


Student registration
Student
Admin
Project registration
Project
Admin
Allocate project
Allocation
Project status
Update status
Student

B. TABLE STRUCTURE

The table needed for each module was 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 NAME: ADMIN

FIELD	DATA TYPE	SIZE	CONSTRAINT
Adminid	Int	10	Primary key
Username	Varchar	20	Not null
password	Varchar	20	Not null

TABLE NAME: STUDENT

FIELD NAME	FIELD TYPE	SIZE	CONSTRAINT
studentid	int	10	Primary key
name	varchar	20	Not null
rollno	varchar	15	Not null
department	varchar	25	Not null
classname	varchar	15	Not null
guidename	varchar	20	Not null
mobile	varchar	10	Not null
username	varchar	20	Not null
password	varchar	20	Not null

TABLE NAME: PROJECT

FILED NAME	FIELD TYPE	SIZE	CONSTRAINT
projectid	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 NAME: ALLOCATE

FIELD NAME	FIELD TYPE	SIZE	CONSTRAINT
id	int	10	Primary key
studentid	int	10	Foreign key
projectid	int	10	Foreign key
status	varchar	30	Not null

C.SAMPLE CODING

```
#!/usr/bin/env python
# package: com.example.demo.controller
import python.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

@RequestMapping(value="/api")
class ApiController(object):
    """ generated source for class ApiController """
    service = ApiService()
    dao = ApiDao()

    @GetMapping("/login/{username}/{password}")
    def login(self, username, password):
        """ generated source for method login """
```

```
        return self.service.login(username, password)


@GetMapping("/add_student/{name}/{rollno}/{department}/{classname}/{guide}/{mobile}/{username}/{password}")
    def member_register(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method member_register """
        self.dao.studentRegister(name, rollno, department, classname, guide, mobile, username, password)
        return "Student Saved Sucessfully"

    @GetMapping("/add_project/{title}/{desc}/{modules}/{technology}")
    def add_project(self, title, desc, modules, technology):
        """ generated source for method add_project """
        self.dao.add_project(title, desc, modules, technology)
        return "Project Created "

    @overloaded
    @GetMapping("/allocate_project/{studentid}/{projectid}")
    def allocate_project(self, studentid, projectid):
        """ generated source for method allocate_project """
        self.dao.allocate_project(studentid, projectid)
        return "Project Allocated Completed Sucessfully"

    @GetMapping("/update_status/{allocate_id}/{status}")
    @allocate_project.register(object, int, str)
    def allocate_project_0(self, allocate_id, status):
        """ generated source for method allocate_project_0 """
        self.dao.update_project(allocate_id, status)
        return "Project Status updated Sucessfully"

    @GetMapping("/get_projects")
    def get_projects(self):
        """ generated source for method get_projects """
        return self.dao.get_projects()

    @GetMapping("/get_projectsA")
    def get_projectsA(self):
        """ generated source for method get_projectsA """
        return self.dao.get_projectsA()
```

```
    @overloaded
    @GetMapping("/get_student")
    def get_student(self):
        """ generated source for method get_student """
        return self.dao.get_student()

    @GetMapping("/get_student/{id}")
    @get_student.register(object, int)
    def get_student_0(self, id):
        """ generated source for method get_student_0 """
        return self.dao.get_student(id)

        #!/usr/bin/env python
# package: com.example.demo.dao
import python.text.DateFormat

import python.text.SimpleDateFormat

import python.util.Date

import python.util.List

import pythonx.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

class ApiDao(object):
    """ generated source for class ApiDao """
    sf = SessionFactory()

    def login(self, username, password):
```

```
        """ generated source for method login """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "select * from admin where username='" + username + "' and password='" + password + "'"
        nq = session.createNativeQuery(sql)
        if nq.list_().size() != 0:
            return "admin"
        else:
            if nq1.list_().size() != 0:
                return "id=" + a.get(0)[0]
            else:
                return "Invalid"

    def studentRegister(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method studentRegister """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "INSERT INTO `student` (`id`, `name`, `rollno`, `department`, `classname`, `guidename`, `mobile`, `username`, `password`) VALUES " + "(NULL, '" + name + "', '" + rollno + "', '" + department + "', '" + classname + "', '" + guide + "', '" + mobile + "', '" + username + "', '" + password + "');"
        print "test" + sql,
        session.createSQLQuery(sql).executeUpdate()

    def add_project(self, title, desc, modules, technology):
        """ generated source for method add_project """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "INSERT INTO `project` (`id`, `title`, `description`, `modules`, `technology`) VALUES " + "(NULL, '          " + title + "', '" + desc + "', '" + modules + "', '" + technology + "');"
        session.createSQLQuery(sql).executeUpdate()

    def allocate_project(self, studentid, projectid):
        """ generated source for method allocate_project """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "INSERT INTO `allocate` (`id`, `studentid`, `projectid`, `status`) VALUES " + "(NULL, '" + studentid + "', '" + projectid + "', 'Initiated');"
        session.createSQLQuery(sql).executeUpdate()

    def get_projects(self):
```

```
        """ generated source for method get_projects """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql="selects.idas 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)"
        nq = session.createNativeQuery(sql)
        return nq.list_()

    @overloaded
    def get_student(self, id):
        """ generated source for method get_student """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()

sql="selects.idasstudent,p.id,s.name,s.rollno,p.title,p.description,a.status,p.modules,p.technology,s.gui
dename,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
        nq = session.createNativeQuery(sql)
        return nq.list_()

    def get_projectsA(self):
        """ generated source for method get_projectsA """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "select * from project"
        nq = session.createNativeQuery(sql)
        return nq.list_()

    @get_student.register(object)
    def get_student_0(self):
        """ generated source for method get_student_0 """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "select * from student"
        nq = session.createNativeQuery(sql)
        return nq.list_()

    def update_project(self, allocate_id, status):
        """ generated source for method update_project """
        #  TODO Auto-generated method stub
```

```
        session = self.sf.getCurrentSession()
        sql = "UPDATE `allocate` SET `status` = '" + status + "' WHERE `allocate`.`id` = " + allocate_id
        session.createSQLQuery(sql).executeUpdate()
```

```
#!/usr/bin/env python
# package: com.example.demo.controller
import python.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

@RequestMapping(value="/api")
class ApiController(object):
    """ generated source for class ApiController """
    service = ApiService()
    dao = ApiDao()

    @GetMapping("/login/{username}/{password}")
    def login(self, username, password):
        """ generated source for method login """
        return self.service.login(username, password)
```

```
@GetMapping("/add_student/{name}/{rollno}/{department}/{classname}/{guide}/{mobile}/{username}/{password}")
    def member_register(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method member_register """
        self.dao.studentRegister(name, rollno, department, classname, guide, mobile, username, password)
        return "Student Saved Sucessfully"

    @GetMapping("/add_project/{title}/{desc}/{modules}/{technology}")
    def add_project(self, title, desc, modules, technology):
        """ generated source for method add_project """
        self.dao.add_project(title, desc, modules, technology)
        return "Project Created "

    @overloaded
    @GetMapping("/allocate_project/{studentid}/{projectid}")
    def allocate_project(self, studentid, projectid):
        """ generated source for method allocate_project """
        self.dao.allocate_project(studentid, projectid)
        return "Project Allocated Completed Sucessfully"

    @GetMapping("/update_status/{allocate_id}/{status}")
    @allocate_project.register(object, int, str)
    def allocate_project_0(self, allocate_id, status):
        """ generated source for method allocate_project_0 """
        self.dao.update_project(allocate_id, status)
        return "Project Status updated Sucessfully"

    @GetMapping("/get_projects")
    def get_projects(self):
        """ generated source for method get_projects """
        return self.dao.get_projects()

    @GetMapping("/get_projectsA")
    def get_projectsA(self):
        """ generated source for method get_projectsA """
        return self.dao.get_projectsA()
```

```
    @overloaded
    @GetMapping("/get_student")
    def get_student(self):
        """ generated source for method get_student """
        return self.dao.get_student()

    @GetMapping("/get_student/{id}")
    @get_student.register(object, int)
    def get_student_0(self, id):
        """ generated source for method get_student_0 """
        return self.dao.get_student(id)

@RequestMapping(value="/api")
class ApiController(object):
    """ generated source for class ApiController """
    service = ApiService()
    dao = ApiDao()

    @GetMapping("/login/{username}/{password}")
    def login(self, username, password):
        """ generated source for method login """
        return self.service.login(username, password)


@GetMapping("/add_student/{name}/{rollno}/{department}/{classname}/{guide}/{mobile}/{userna
me}/{password}")
    def member_register(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method member_register """
        self.dao.studentRegister(name, rollno, department, classname, guide, mobile, username,
password)
        return "Student Saved Sucessfully"

    @GetMapping("/add_project/{title}/{desc}/{modules}/{technology}")
    def add_project(self, title, desc, modules, technology):
        """ generated source for method add_project """
        self.dao.add_project(title, desc, modules, technology)
        return "Project Created "

    @overloaded
    @GetMapping("/allocate_project/{studentid}/{projectid}")
```

```
    def allocate_project(self, studentid, projectid):
        """ generated source for method allocate_project """
        self.dao.allocate_project(studentid, projectid)
        return "Project Allocated Completed Sucessfully"

    @GetMapping("/update_status/{allocate_id}/{status}")
    @allocate_project.register(object, int, str)
    def allocate_project_0(self, allocate_id, status):
        """ generated source for method allocate_project_0 """
        self.dao.update_project(allocate_id, status)
        return "Project Status updated Sucessfully"

    @GetMapping("/get_projects")
    def get_projects(self):
        """ generated source for method get_projects """
        return self.dao.get_projects()

    @GetMapping("/get_projectsA")
    def get_projectsA(self):
        """ generated source for method get_projectsA """
        return self.dao.get_projectsA()

    @overloaded
    @GetMapping("/get_student")
    def get_student(self):
        """ generated source for method get_student """
        return self.dao.get_student()

    @GetMapping("/get_student/{id}")
    @get_student.register(object, int)
    def get_student_0(self, id):
        """ generated source for method get_student_0 """
        return self.dao.get_student(id)

        #!/usr/bin/env python
# package: com.example.demo.configuration
import python.util.Properties

import pythonx.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.StudentRegistration

import org.springframework.orm.Admin.Allocate.content

import org.springframework.orm.Update.annotation.process

import org.springframework.Student.project.Tracking.management

    def studentRegister(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method studentRegister """
        #  TODO Auto-generated method stub
        session = self.sf.getCurrentSession()
        sql = "INSERT INTO `student` (`id`, `name`, `rollno`, `department`, `classname`, `guidename`, `mobile`, `username`, `password`) VALUES " + "(NULL, '" + name + "', '" + rollno + "', '" + department + "', '" + classname + "', '" + guide + "', '" + mobile + "', '" + username + "', '" + password + "');"
        print "test" + sql,
        session.createSQLQuery(sql).executeUpdate()
@GetMapping("/add_student/{name}/{rollno}/{department}/{classname}/{guide}/{mobile}/{username}/{password}")
    def member_register(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method member_register """
        self.dao.studentRegister(name, rollno, department, classname, guide, mobile, username, password)
        return "Student Saved Sucessfully"

    @GetMapping("/add_project/{title}/{desc}/{modules}/{technology}")
    def add_project(self, title, desc, modules, technology):
        """ generated source for method add_project """
        self.dao.add_project(title, desc, modules, technology)
        return "Project Created "

    @overloaded
    @GetMapping("/allocate_project/{studentid}/{projectid}")
    def allocate_project(self, studentid, projectid):
```

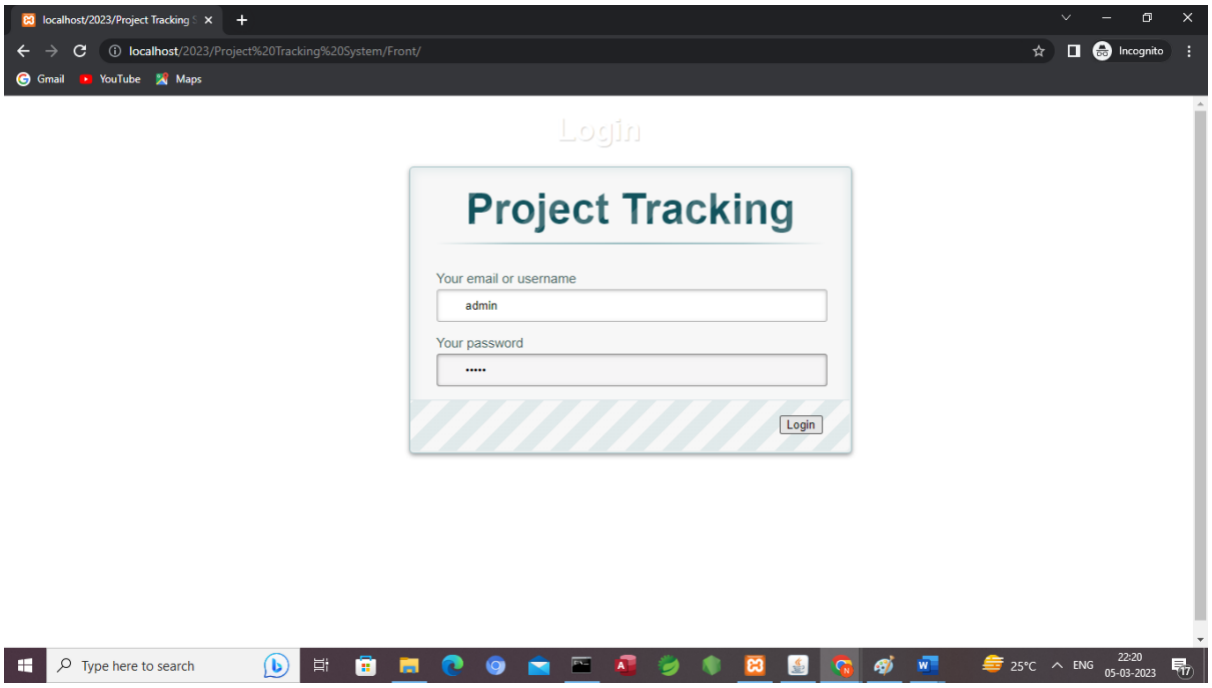
```
        """ generated source for method allocate_project """
        self.dao.allocate_project(studentid, projectid)
        return "Project Allocated Completed Sucessfully"

    @GetMapping("/update_status/{allocate_id}/{status}")
    @allocate_project.register(object, int, str)
    def allocate_project_0(self, allocate_id, status):
        """ generated source for method allocate_project_0 """
        self.dao.update_project(allocate_id, status)
        return "Project Status updated Sucessfully"

    @GetMapping("/get_projects")
    def get_projects(self):
        """ generated source for method get_projects """
        return self.dao.get_projects()

    @GetMapping("/get_projectsA")
    def get_projectsA(self):
        """ generated source for method get_projectsA """
        return self.dao.get_projectsA()

#!/usr/bin/env python
# package: com.example.demo.controller
import python.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

@RequestMapping(value="/api")
class ApiController(object):
    """ generated source for class ApiController """
    service = ApiService()
    dao = ApiDao()

    @GetMapping("/login/{username}/{password}")
    def login(self, username, password):
        """ generated source for method login """
        return self.service.login(username, password)


@GetMapping("/add_student/{name}/{rollno}/{department}/{classname}/{guide}/{mobile}/{username}/{password}")
    def member_register(self, name, rollno, department, classname, guide, mobile, username, password):
        """ generated source for method member_register """
        self.dao.studentRegister(name, rollno, department, classname, guide, mobile, username, password)
        return "Student Saved Sucessfully"

    @GetMapping("/add_project/{title}/{desc}/{modules}/{technology}")
    def add_project(self, title, desc, modules, technology):
        """ generated source for method add_project """
        self.dao.add_project(title, desc, modules, technology)
        return "Project Created "

    @overloaded
    @GetMapping("/allocate_project/{studentid}/{projectid}")
    def allocate_project(self, studentid, projectid):
        """ generated source for method allocate_project """
        self.dao.allocate_project(studentid, projectid)
        return "Project Allocated Completed Sucessfully"
```

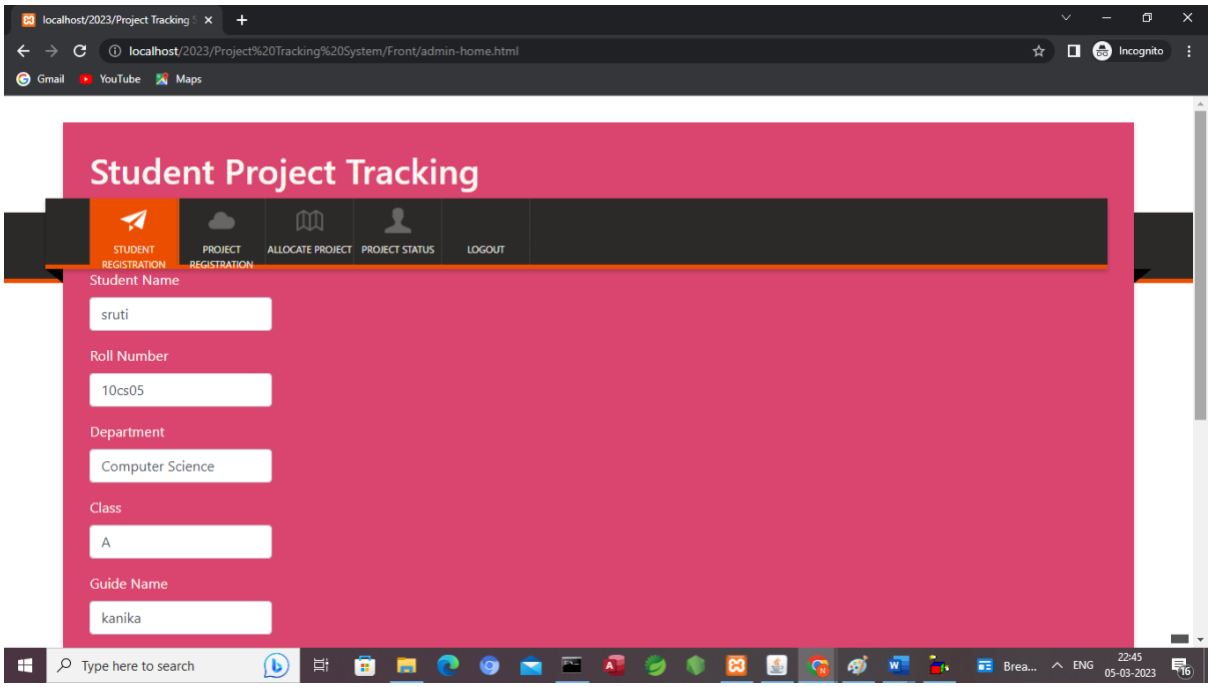
```
@GetMapping("/update_status/{allocate_id}/{status}")
@allocate_project.register(object, int, str)
def allocate_project_0(self, allocate_id, status):
    """ generated source for method allocate_project_0 """
    self.dao.update_project(allocate_id, status)
    return "Project Status updated Sucessfully"

@GetMapping("/get_projects")
def get_projects(self):
    """ generated source for method get_projects """
    return self.dao.get_projects()

@GetMapping("/get_projectsA")
def get_projectsA(self):
    """ generated source for method get_projectsA """
    return self.dao.get_projectsA()

@overloaded
@GetMapping("/get_student")
def get_student(self):
    """ generated source for method get_student """
    return self.dao.get_student()

@GetMapping("/get_student/{id}")
@get_student.register(object, int)
def get_student_0(self, id):
    """ generated source for method get_student_0 """
    return self.dao.get_student(id)
```

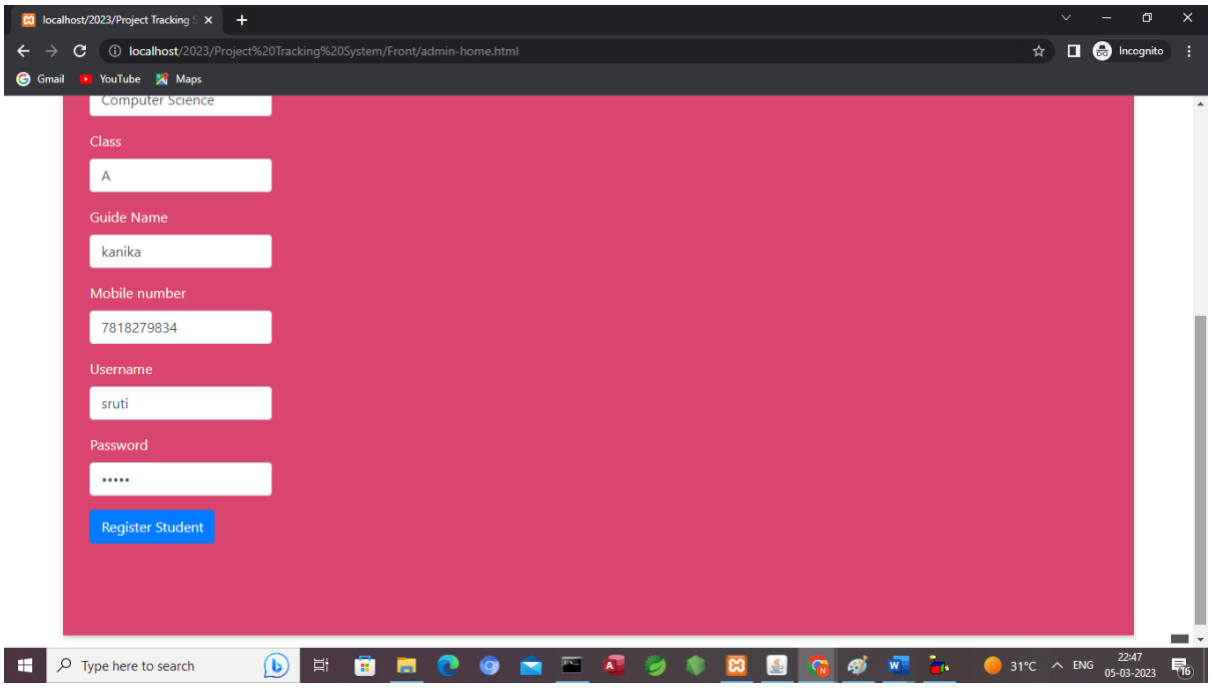
D.SAMPLE INPUT & OUTPUT DESIGN

Login page (Admin login)

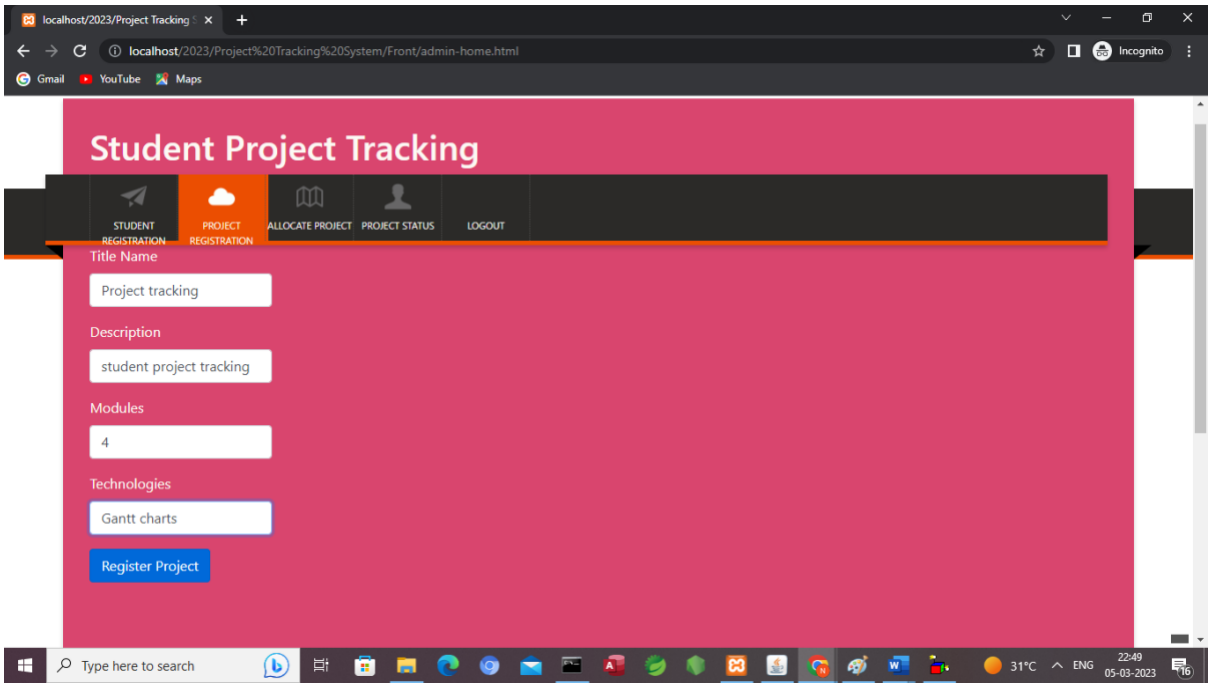


Student registration

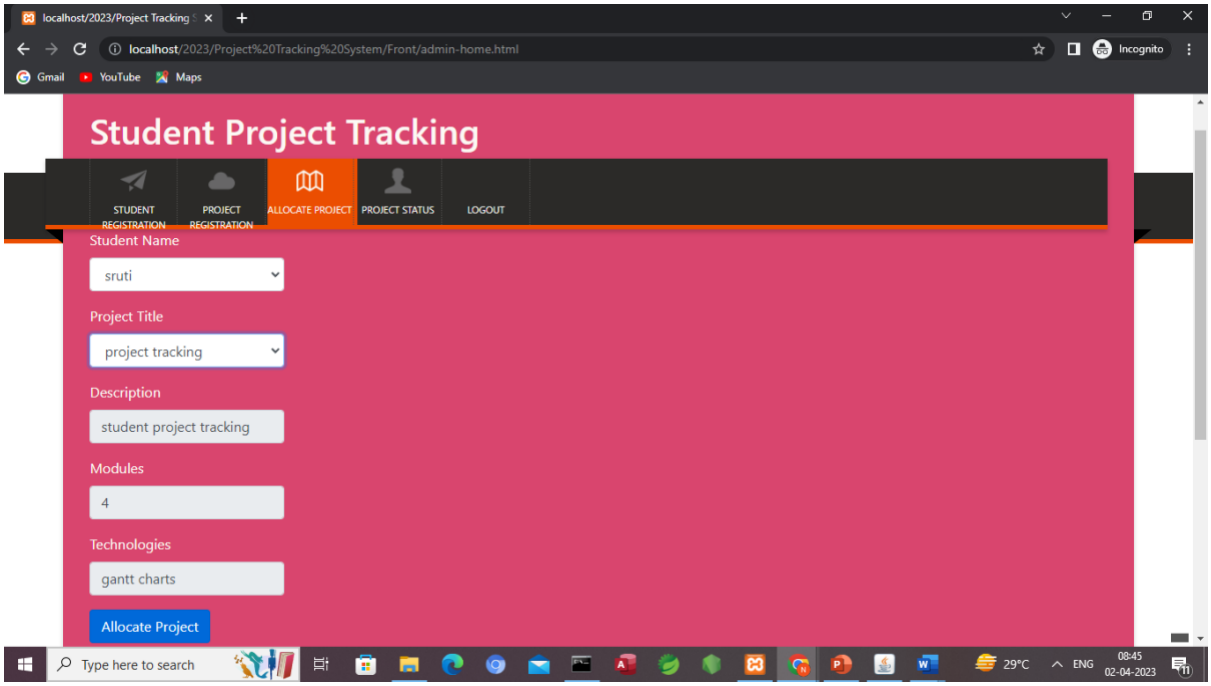




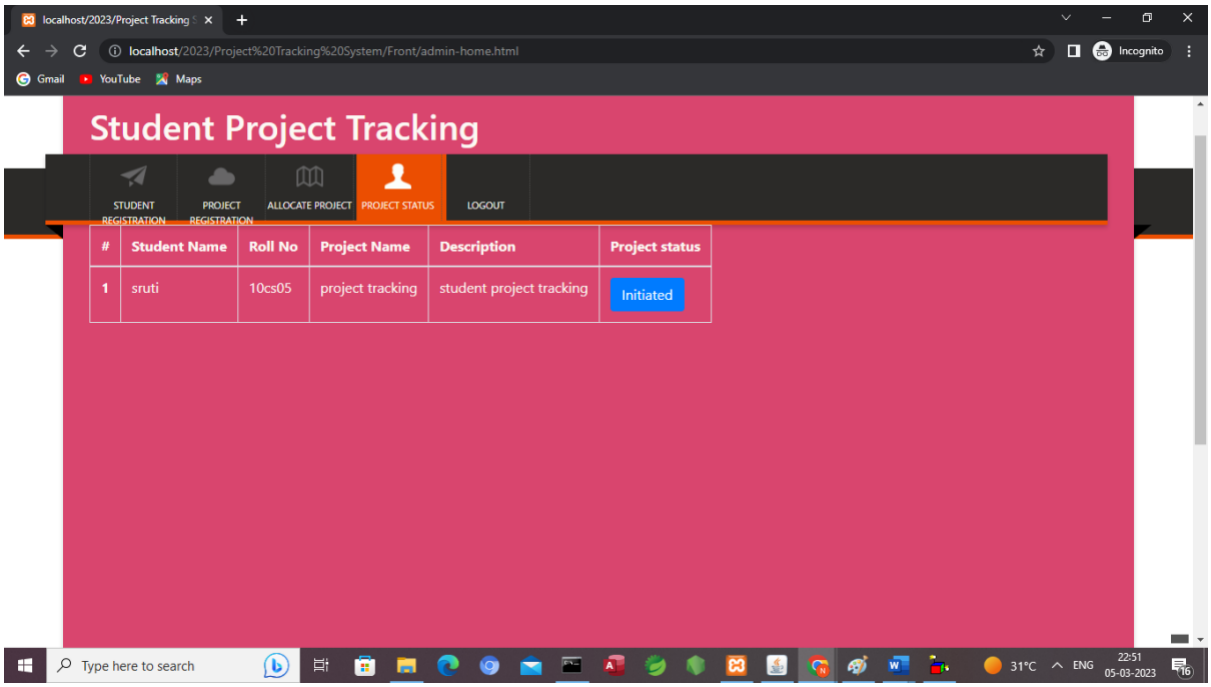
Project registration



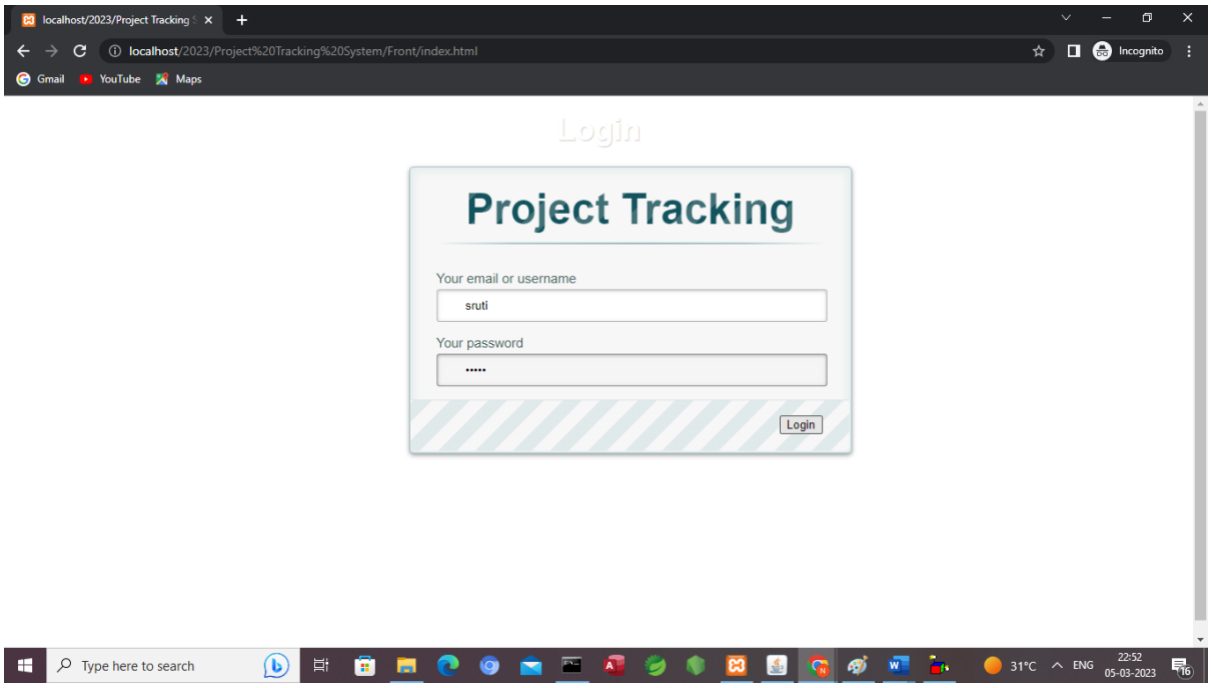
Allocate project



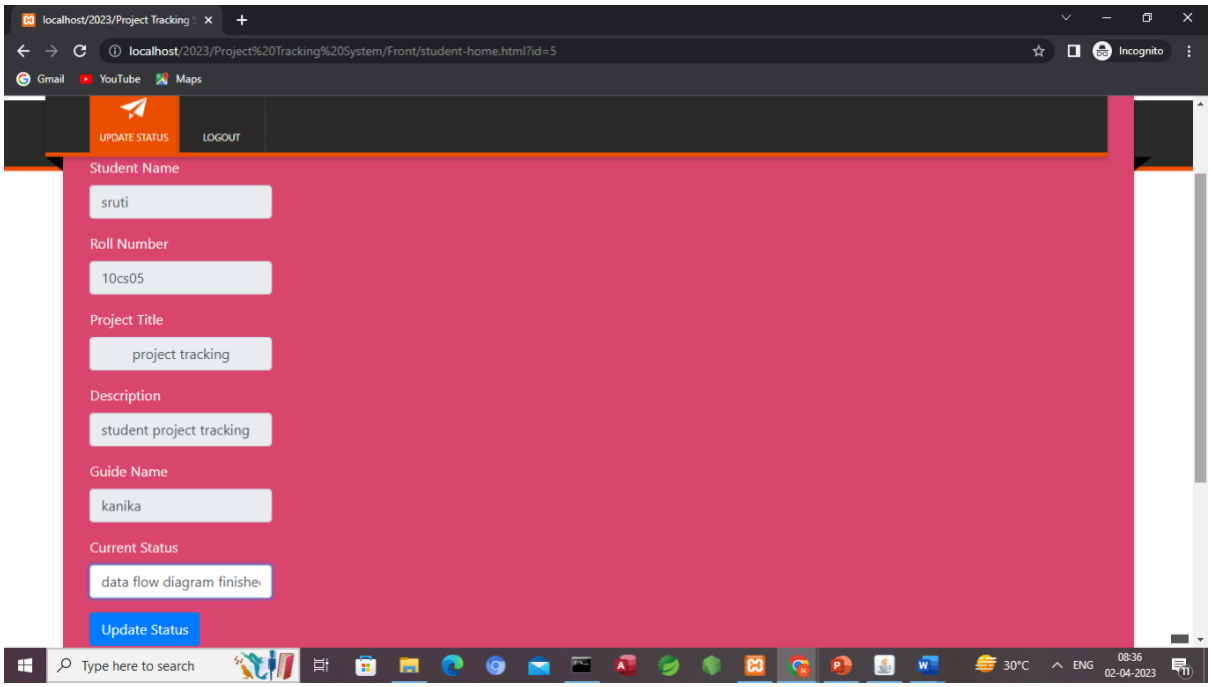
Project status



Login page(Student login)



Update status



Project status