

System Design Storage Document

**Table of Contents**

[1 Introduction 3](#_Toc287863394)

[1.1 Purpose of Document 3](#_Toc287863395)

[1.2 Document Scope 3](#_Toc287863396)

[1.2.1 In-Scope 4](#_Toc287863397)

[1.2.2 Out-of-Scope 4](#_Toc287863398)

[1.2.3 Assumptions 4](#_Toc287863399)

[1.3 Methodology and Tools 4](#_Toc287863400)

[2 Design Overview 5](#_Toc287863402)

[2.1 Background Information 5](#_Toc287863403)

[2.2 System Evolution Description 5](#_Toc287863404)

[2.3 Required Environment 6](#_Toc287863405)

[2.4 Constraints 7](#_Toc287863406)

[2.5 Design Trade-offs 7](#_Toc287863407)

[3 Logical Architecture 7](#_Toc287863408)

[3.1 Application Architecture 7](#_Toc287863410)

[3.2 Communication Architecture 7](#_Toc287863411)

[4 Data Model 8](#_Toc287863413)

[4.1 Database Management System Files 8](#_Toc287863414)

[4.2 Non-Database Management System Files 9](#_Toc287863415)

[5 Detailed Design 9](#_Toc287863416)

[5.1 Application Detailed Design 9](#_Toc287863418)

[5.2 Communication Detailed Design 11](#_Toc287863419)

[6 Graphical User Interface (GUI) 12](#_Toc287863423)

[6.1 Navigation Hierarchy 12](#_Toc287863427)

# Introduction

The University Project Bank, namely ‘Anusandhaan’, is a software system designed to enable university students to create accounts, view various projects, and submit their own project proposals for review by an administrator. The administrator can accept or reject the proposals, and a token system is used to track the status of each proposal. This document outlines the functional and non-functional requirements of the system, as well as the system architecture, design, and implementation details.

## Purpose of Document

The document provides comprehensive details with which the software has been developed. It contains detailed information regarding the functionalities of the software and the way it has been implemented.

## Document Scope

The following document is expected to give a detailed description of the system requirements, operating environment, architecture, files, and database design of the software design implemented.

### In-Scope

### User account creation, login, and logout

### Project browsing and searching.

### Project proposal submission with title, description, and associated files

### Project proposal review by Admin.

### Project proposal status tracking using a unique token system.

### Administrator ability to accept or reject project proposals.

### Out-of-Scope

* Payment processing for projects.
* Integration with external services or APIs.
* Advanced project management features such as project timelines or task assignment.
* Automated project matching or recommendation systems.
* Social features such as user-to-user messaging or forums.
* Mobile app development or support.

### Assumptions

* All users have a reliable internet connection and access to a modern web browser.
* The system will be used primarily by university students and administrators.
* Users will provide accurate and complete information when creating accounts and submitting project proposals.
* Administrators will review project proposals in a timely manner.

## Methodology and Tools

The software has been produced by implementing the incremental methodology. It has been built and delivered in small increments, each of which adds new features or functionality. Each increment is developed, tested, and delivered independently, and the final product is created by integrating all the increments. The requirements were divided into small chunks, and each chunk was developed and tested before moving on to the next chunk. This approach allowed feedback on the software at each stage of the development procedure, and to make changes or improvements based on that feedback making it more convenient to detect flaws during execution and testing of each chunk.

The tools used for implementing this software are PHP and MySQL. PHP has been used as it is easy to learn and implement, open source and due to simpler database connectivity. MySQL has been used as it is easy to learn and implement, open source and due to its compatibility with PHP.

# Design Overview

## Background Information

The key user groups for the Anusandhaan project include students and admin. Students will be the primary users of the system and will be able to create an account, browse projects, and submit publishing requests. Admin will be responsible for managing the projects, deciding whether to accept or reject publishing requests, and will be responsible for maintaining the system and providing support to users as needed.

## System Evolution Description

The development of the website had been carried out by dividing it into various subtasks. The subtasks are listed below:

* **Student Sign Up / Sign In:** It has been made mandatory to use an account for accessing any ***Student*** functionality like viewing approved projects, to submit their own project, etc.
* **Admin Sign Up / Sign In:** It has been made mandatory to use an account for accessing any ***Admin*** functionality like checking pending project requests, accepting / rejecting projects, etc.
* **Upload Project:** This functionality has been made available only for the ***Students***wherein they could submit their project with the appropriate details mentioned.
* **Token Generation:** This functionality would be automatically generated which would help to keep track of various project requests received from various students for Admin. From Students point of view, it helps them to track their various project submissions to view its status.
* **Project Approval:** This functionality has been made available only for the ***Admin*** wherein they could approve / reject the various project requests that have been received from the students.
* **Project Status:** This functionality has been made available only for the ***Students*** wherein a student could view their uploaded project requests’ status appropriately.
* **Search Project:** This functionality has been made available to both the ***students*** and the ***admin***. Keywords can be inserted into the search bar to search for the users’ desired results.

## Required Environment

|  |  |
| --- | --- |
| Functionality | Requirements |
| Operating System | * Windows * Linux * macOS |
| Web and Database Server | * XAMPP (PHP and MySQL) * JavaScript |
| Development Tools | * Visual Studio Code |

## Constraints

* It is assumed that the admin responds to the various project requests in the appropriate time interval as there is no auto accept / reject of projects.
* It is assumed that the Students and the Admin are aware about the token generation system and its working.
* There is no built-in plagiarism checker and would need to be carried out using third-party applications whenever necessary.
* Search Engine Optimization could be implemented for better searching accuracy.

## Design Trade-offs

The Design Trade-offs are mentioned as follows:

1. Innovation vs Practicality: It faces a trade-off between innovation and practicality. While it may be tempting to incorporate the latest technologies and features, doing could prove challenging to implement it righteously by the specified deadline.
2. Security vs Usability: A trade-off is the balance between security and usability. While it's important to ensure that the platform is secure, too many security measures may make it difficult for users to access and use the system.

# Logical Architecture

## Application Architecture

It has been implemented using PHP and MySQL. PHP is responsible for processing user requests, accessing the database, and generating appropriate responses whereas MySQL is responsible for storing the data by providing the database.

## Communication Architecture

It will use the client-server communication architecture this is because when a user interacts with the system, their request will be sent to the server, which will then process the request and then send a response back to the client who had sent the request.

# Data Model

## Database Management System Files

The tables created for Database Management are as follows:

1. Users

* id
* name
* email
* password
* contact\_number

1. Admins

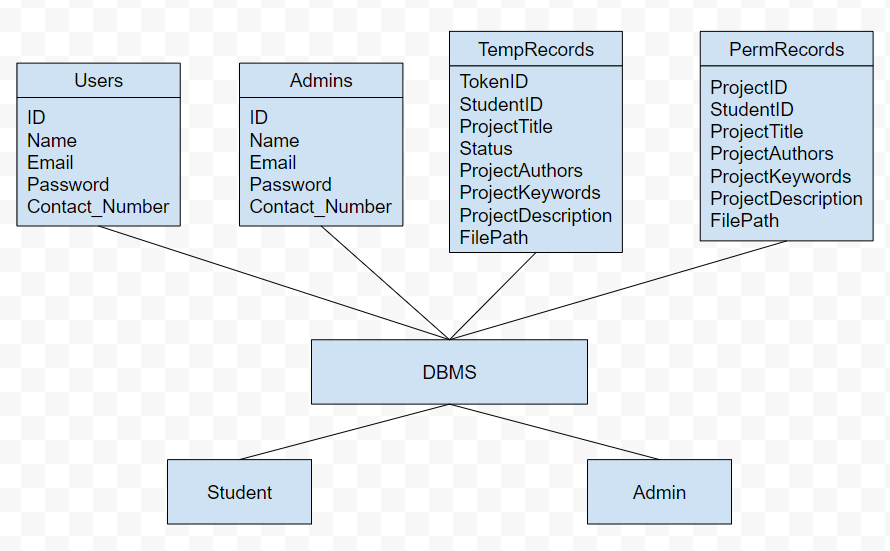
* ID
* Name
* email
* password
* contact\_number

1. TempRecords

* tokenID
* studentID
* projectTitle
* status
* projectAuthors
* projectKeywords
* projectDescription
* filePath

1. PermRecords

* projectID
* studentID
* projectTitle
* projectAuthors
* projectKeywords
* projectDescription
* filePath



## Non-Database Management System Files

The PDFs that have been uploaded by the students during project submission is stored as .pdf, and is downloaded in the ‘upload’ folder of the server system. Any student could download on their local system in their desired file location.

# Detailed Design

## Application Detailed Design

The application design is intended to provide a user-friendly interface which would allow students to easily access the projects via search and view their publication requests’ status while also providing admins with the necessary tools to manage the system. The design implemented is as follows:

1. Login and Registration Module: This module will allow users to create an account in the system and log in using their credentials. It will consist of the following components:

* Registration Form: A web form that allows users to enter their name, email address, password, and other information required to create account.
* Login Form: A web form that allows users to enter their credentials and log in to their account if valid credentials are entered.

1. Project Management Module: This module will allow users to view and manage projects available in the Anusandhaan system. It will consist of the following components:

* Project Listing: A web page that lists all the available projects with details like name, description, start and end dates, and status.
* Project Details Page: A web page that displays detailed information about a selected project, including its associated user ID.
* Project Creation Form: A web form that allows users to create new projects and add relevant details.

1. Publication Request Management Module: This module will allow users to request publication of their projects in the Anusandhaan system. It will consist of the following components:

* Publication Request Form: A web form that allows users to submit a publication request for their project, including details like associated project ID and request date.
* Publication Request Listing: A web page that lists all the publication requests made by the user with details like its status and associated token value.

1. Admin Module: This module will allow administrators to manage projects, and publication requests in the Anusandhaan system. It will consist of the following components:

* Project Management: A web page that lists all the available projects with details like name, description, and status.
* Publication Request Management: A web page that lists all the publication requests made by users with details like request date, status, and associated token value. Admins can manage publication requests by updating their status or rejecting them.

1. Token Management Module: This module will manage the auto-generated tokens associated with each publication request made by the user. It will consist of the following components:

* Token Generation Service: A service that generates unique tokens for each publication request made by the user and associates them with the request ID.
* Token Verification Service: A service that verifies the validity of the token provided by the user before approving the publication request.

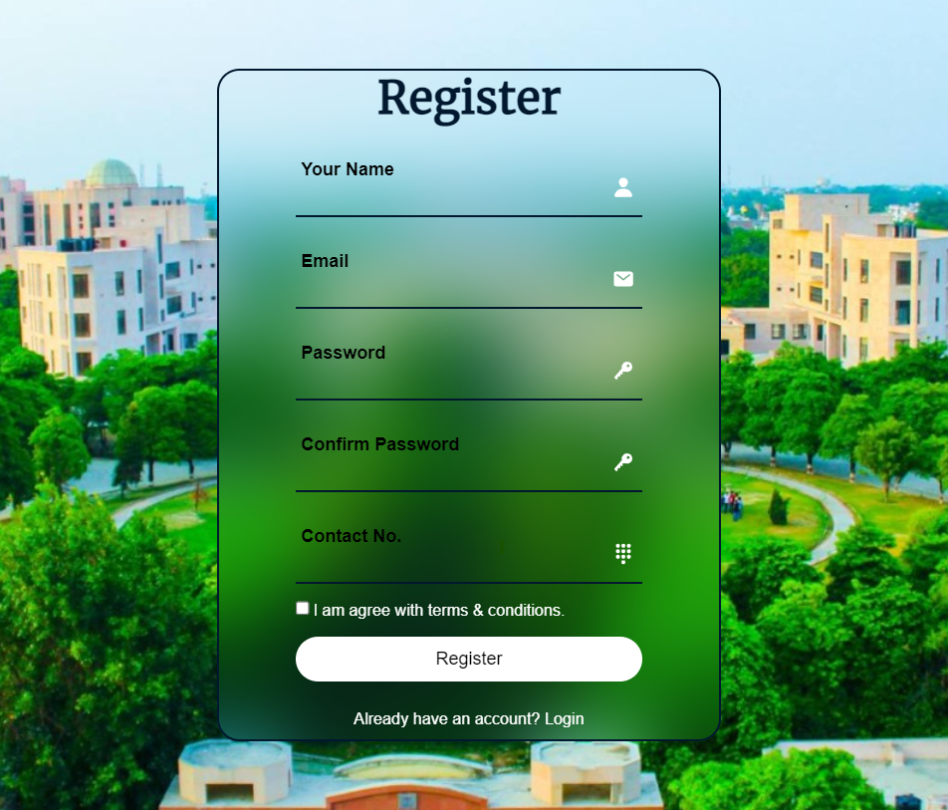
## Communication Detailed Design

* **Client-Server Communication:** Anusandhaan is a web-based system that requires communication between the client and the server. The client communicates with the server. The server responds to the client requests and sends the appropriate data back to the client. The communication between the client and server should be made secure.
* **Server-Server Communication:** Anusandhaan system components such as the web server, application server, and database server communicate with each other. The communication between these servers should be secure and reliable to ensure that the system functions correctly.
* **Email Communication:** Anusandhaan system sends email notifications to users such as project publish requests or acceptance/rejection notifications.

# Graphical User Interface

## Navigation Hierarchy

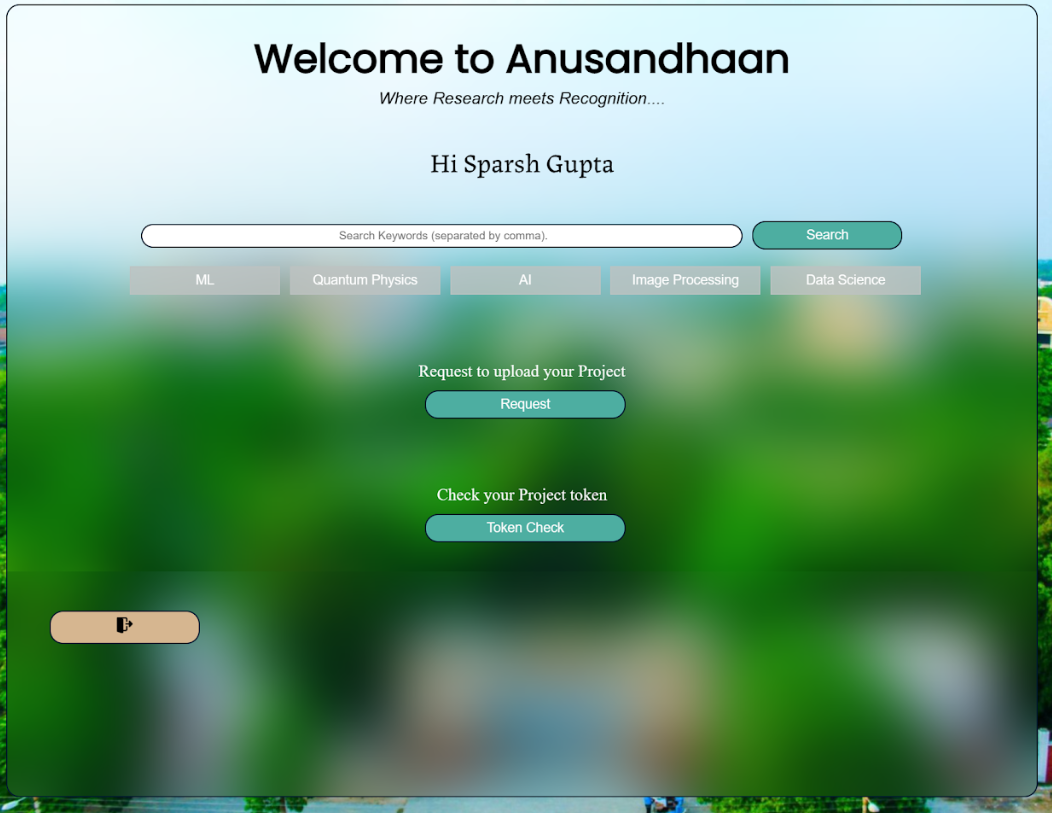
### StudentSignUpPage



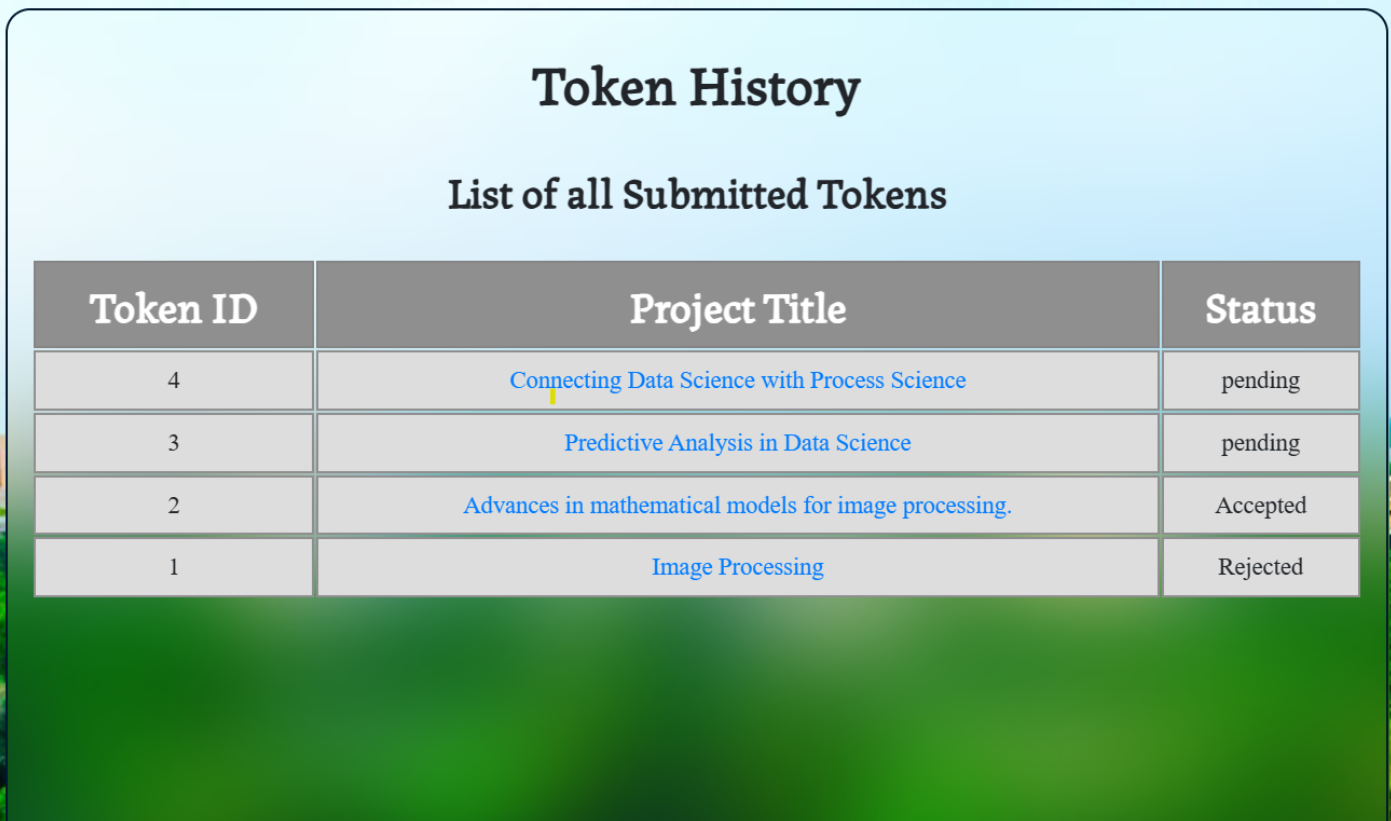
### StudentSignInPage



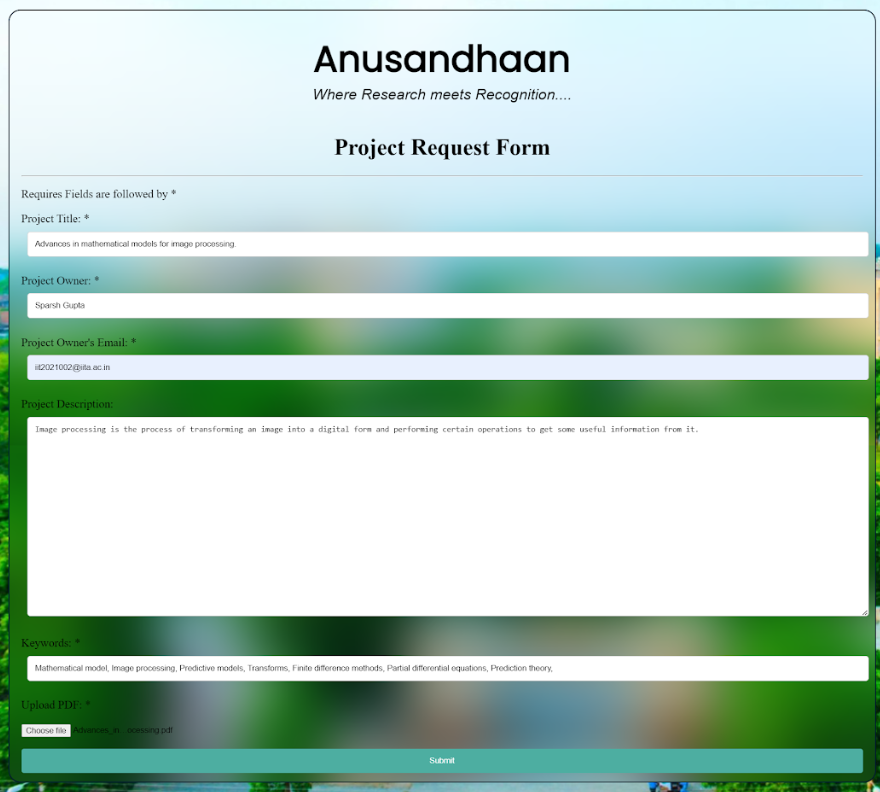
### StudentHomePage

****

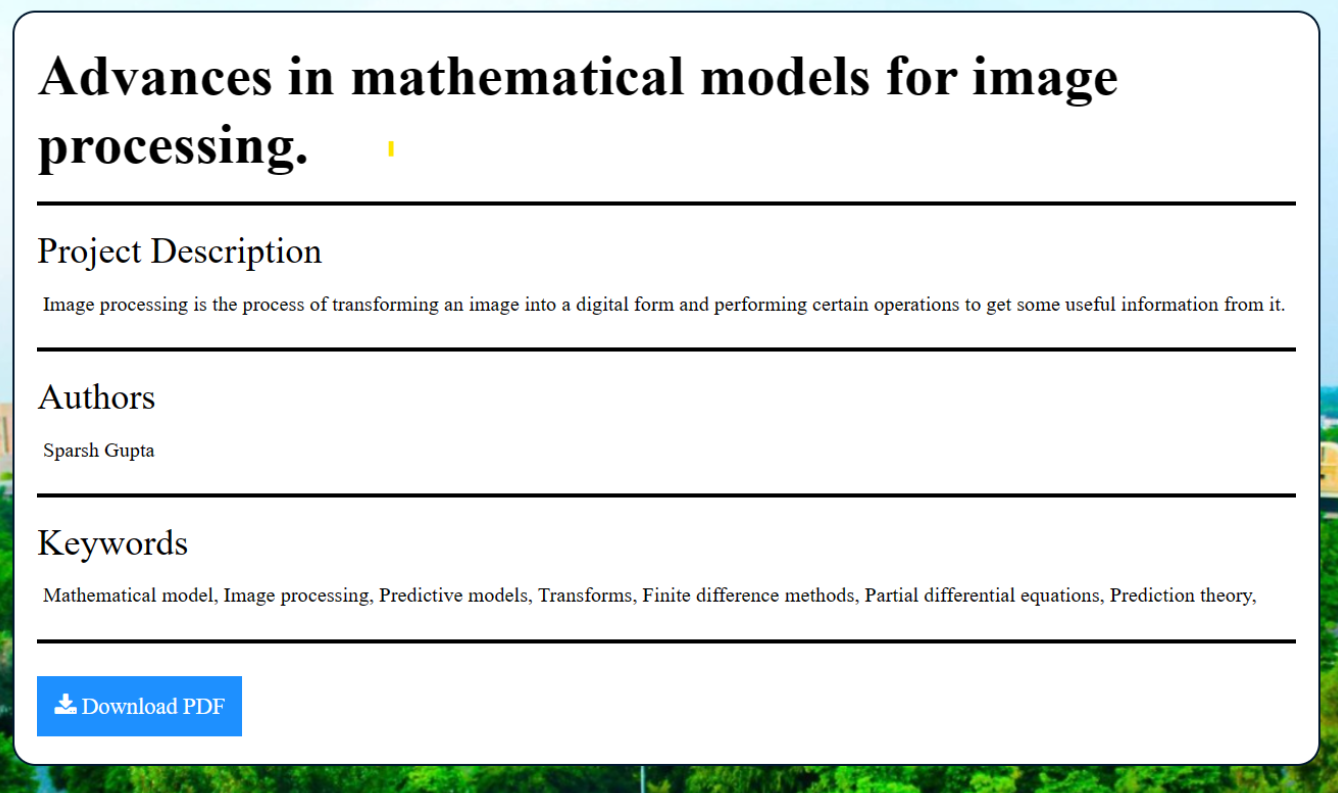
### StudentTokenHistory

****

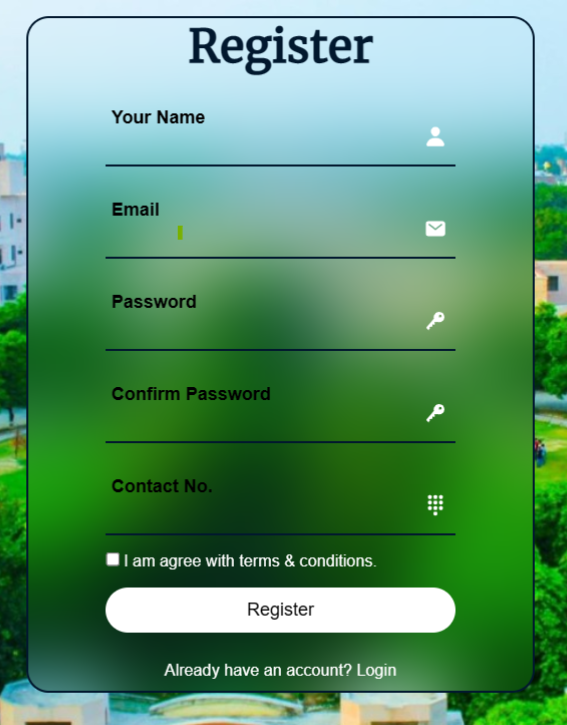
### StudentProjectUpload

****

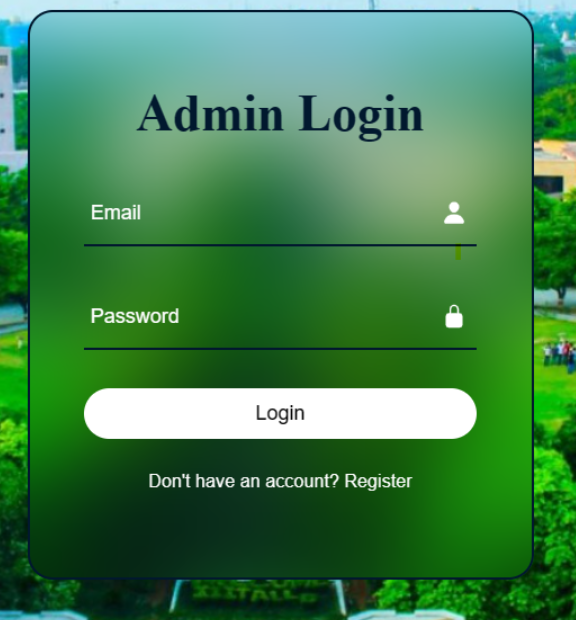
### StudentProjectPage

****

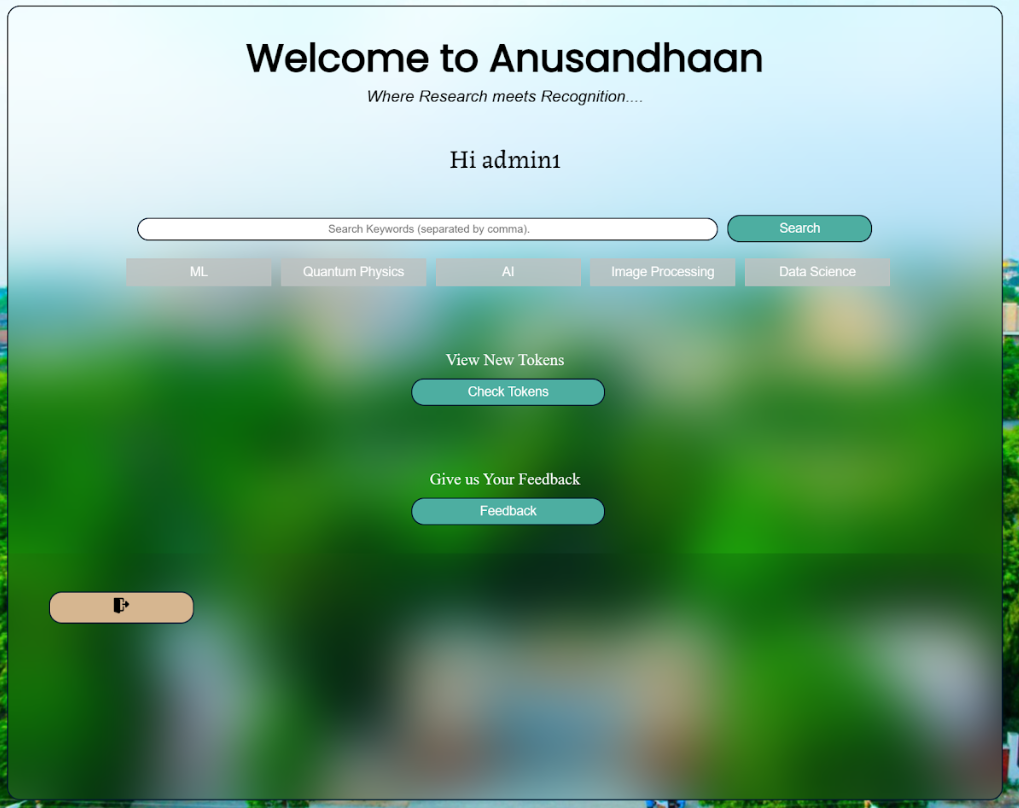
### AdminRegisterPage

****

### AdminSignInPage

****

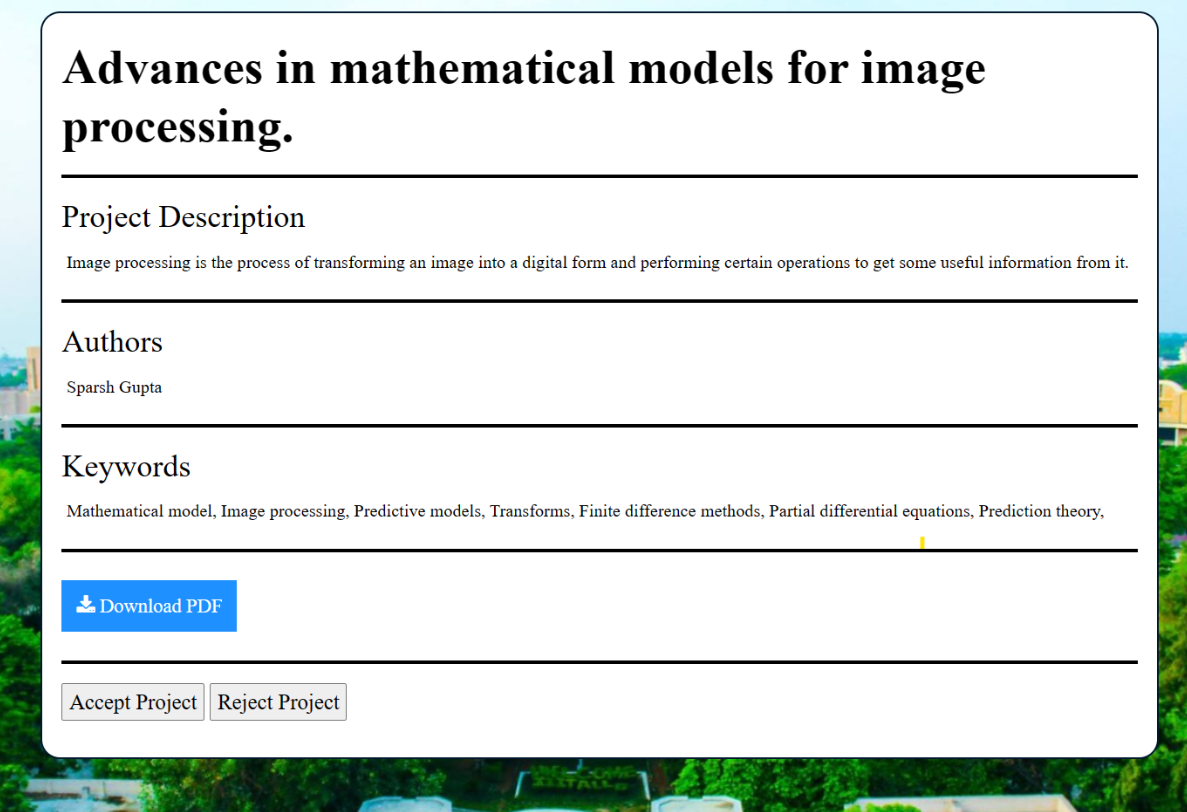
### AdminWelcomePage

****

### AdminCheckTokenPage

****

### AdminPendingActionPage

****