SCU
SLIIT CITY UNI

Final Project Report

for

SLIITNU Panel

By

TechNUOB

Client Northern Uni

Version 1.0

Prepared by

TechNUOB

Declaration

We hereby declare that the project work entitled "**SLIITNU Panel**", submitted to the SLIIT CITY UNI (Pvt.) Ltd. is a record of an original work done by us, under the guidance of our Supervisor **Mr. Kanagasabai Thiruthanigesan**. This project work is submitted in the partial fulfillment of the requirement for the award of the Diploma in Information Technology. The Results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Date of Submission: 6 June 2025

Name of the supervisor:

Mr. Kanagasabai Thiruthanigesan

Acknowledgement

We would like to thank everyone who helped and supported us during our PPA module project. Our project was the SLIITNU Panel**,** a web based system for managing administration tasks at SLIIT Northern Uni**.**

First, we want to give special thanks to Mr. Kanagasabai Thiruthanigesan and Mr. Saroon Mohammed Aflal. They gave us great advice, support, and encouragement throughout the project. Their ideas helped us complete our work in the right way.

We also thank our client, the administration team at Northern Uni**,** for clearly explaining what they needed and giving us helpful feedback. Their support helped us build a system that really matches their needs.

A big thank you goes to our team members. Everyone worked hard, shared ideas, and stayed committed from the beginning to the end of the project. Because of this, we were able to finish the project successfully.

Finally, we thank everyone who helped us, even in small ways by answering our questions, giving support, or helping us solve problems. Your help made a big difference.

Table of Contents

Declaration ii

Acknowledgement iii

Table of Figures v

Group Members vi

1 Executive Summary 1

2 Project Background / Literature Review 2

2.1 Introduction / Background 2

2.2 Similar System / Current Process 3

2.2.1 Manual Administration Management system 3

2.2.2 Current Web System 4

2.3 Proposed Solution 4

3 Problem Definition and Requirement Analysis 5

3.1 Problem Definition 5

3.1.1 People Affected 5

3.1.2 Limitations of Current Manual System 6

3.2 Requirement Analysis 8

3.2.1 Gathering Requirements 8

3.2.2 General Objectives 9

3.2.3 Specified Objectives 10

3.3 Results 11

4 Design and Implementation 20

4.1 High Level Architecture Design 20

4.2 Hardware, Software and Communication Interfaces 21

4.2.1 Hardware Requirements 21

4.2.2 Software Requirements 21

4.2.3 Communication Interfaces 21

4.3 Non-Functional Requirements 22
4.3.1 Operational Requirements 22
4.3.2 Performance Requirements 22
4.3.3 Security Requirements 22
4.4 Implementation 23
4.4.1 Development Methodology 23
4.4.2 Development Technologies 24
4.4.3 Development Tools 25
5 Testing and Deployment 26
6 Future Enhancement 30
7 Summary and Discussion 31
References 32

Table of Figures

Figure 4.1: High Level Architecture Diagram of the System 20
Figure 4.2: Development Methodology Agile Methodology [4] 23

Group Members

Name	**Role**	**Responsibilities**
R. Paviththanan	Backend developer and Project Manager	Backend developer for admin pages, developed email and Telegram notifications, handled all validation for admin pages, responsible for project management.
U. Anojan	Frontend and Backend Developer	Developed and validated the entire enrollment page, implemented a chatbot, completed all report work for the project, and contributed to admin backend functionalities.
R. Abishan	Frontend and Backend Developer	Developed and validated the document upload page and aptitude page, and contributed to admin backend functionalities
S. Janakan	Backend Developer and Tester	Developed and validated the login and signup pages, contributed to admin backend functionalities, and served as the tester for the entire project.
R. Renujan	Frontend and Backend Developer	Developed and validated the program details page, was responsible for database connection and validation, and contributed to admin backend functionalities.
V. Lambodaran	Frontend and Backend Developer	Developed and validated the student dashboard and contributed to admin backend functionalities.

1 Executive Summary

Argument

Our project is called *SLIITNU Panel*. It is a web-based application developed for SLIIT Northern Uni. At the moment, the university uses Excel sheets, notebooks, and physical files to manage student information, payments, and enrollment. This method is time consuming and hard to organize. So, we are creating this web application to make the work easier, faster, and more efficient.

Abstract

In this system, students can complete their enrollment online by submitting the enrollment form and uploading their documents after staff verification. They can also register for their aptitude tests and finish their registration. Once enrolled, students will get access to their own dashboard. From there, they can view their payment history, upload payment slips, raise tickets, update their placements, check announcements, and manage their profiles. A built-in chatbot, Northern Uni's Virtual Assist**,** is also available to help students with questions about courses and related topics, especially when administrative staff are unavailable.

Problem Solution

Administrators will also have their own dashboard. They can manage student details, control access based on staff roles, and download all data as Excel, PDF, or CSV files. Tasks that were done manually before can now be handled easily through this web application.

Conclusion

We are using Django and Wagtail for the backend, and Tailwind CSS, HTML, and JavaScript for the frontend. Currently, the database is SQLite3, but in the future, the system can be migrated to PostgreSQL or another database as needed.

Recommendation

The *SLIITNU Panel* will help the university handle its administration work in a faster, safer, and more organized way. We recommend adding features like automated email notifications and upgrading to a more powerful database when moving to real-time usage.

2 Project Background / Literature Review

SLIIT Northern Uni has been managing its administration work using manual methods like notebooks, Excel sheets, and physical files. This made it hard to manage student data, process registrations, track payments, and maintain records efficiently. So, we decided to build a new web-based application called *SLIITNU Panel*. This system is designed to make the administrative tasks easier by digitalizing the whole process. It allows students to complete their registration online and helps admins to manage all related activities from a central platform.

Before starting this project, we had no experience with Django or other related frameworks. So, we began studying the Django web framework to understand how to use it effectively for our system. We referred to the book *Django by Example* [1], which helped us build a strong foundation in web application development. To handle content management in the admin panel, we explored Wagtail CMS using the guide *Learning Wagtail* [2]. Additionally, to understand how to structure and manage our database, we used *Database System Concepts* [3]. We also referred to several online learning videos to improve our knowledge in Django before starting this project. These resources supported us in learning the necessary tools and concepts for designing and building this project successfully.

2.1 Introduction / Background

SLIIT Northern Uni currently manages its administrative tasks through manual methods like notebooks, Excel sheets, and physical files. This approach is time-consuming, difficult to maintain, and not suitable for growing student and staff numbers. To solve this issue, we are developing a web-based application called *SLIITNU Panel*.

This system is designed to handle key administrative processes such as student enrollment, document verification, aptitude tests, and registration. Once students are registered, they can access a personal dashboard to manage their academic and payment details. At the same time, administrators can use their own dashboard to manage student records, control access based on staff roles, and export data in useful formats like Excel, PDF, or CSV.

Our goal is to provide a faster, safer, and more efficient way for the university to manage its day-to-day administrative work using modern digital tools.

2.2 Similar System / Current Process

SLIIT Northern Uni is currently managing most of its administration work manually through notebooks, Excel files, and printed documents. Although they have an official university website, it only supports limited functions like providing course details, news updates, registration steps, and aptitude test information. That website was mainly created for public access and student use not for administrative purposes. It does not allow admins to log in, handle student records, or perform backend operations like document verification and staff role management.

Because of this, we are developing a new web application called *SLIITNU Panel*, fully focused on administrative needs. In this system, admins will be able to manage student enrollment, handle document checking, update payment details, and download reports easily. We are also including a chatbot feature. This will help students ask questions about courses or registration without needing to call or visit the university. Even though the details are already on the main website, many students still call the administration for extra information. With the chatbot, they can get quick answers, and only contact staff if needed later. This makes the communication process smoother for both sides.

2.2.1 Manual Administration Management system

SLIIT Northern Uni is still handling most of its administration work manually. Staff members record student details, payment history, and other important information using notebooks, Excel sheets, and printed documents. This method is slow and hard to manage, especially when more students are enrolling every year. It also increases the chances of human error, and finding past records takes a lot of time.

2.2.2 Current Web System

The university has an official website which helps students to view course details, register for programs, and see aptitude test information. However, this website is mainly made for student use, not for admin control. It does not allow admin staff to manage student records, verify documents, or control staff access levels. Because of this, we decided to create a new system *SLIITNU Panel* that gives full control to the admin team and supports all backend processes in one platform.

2.3 Proposed Solution

We discussed with university admin staff to understand how they were handling things manually and what features they needed in a digital system. After collecting their feedback, we designed *SLIITNU Panel* to meet all their needs. We showed the early versions of the system to them and made changes based on their suggestions. For example, we updated the design, changed some functions, and added new features when they asked. The final system was created by fully following their requirements and giving them a smooth and easy-to-use platform.

3 Problem Definition and Requirement Analysis

3.1 Problem Definition

SLIIT Northern Uni has been handling its administrative work using manual methods such as notebooks and Excel sheets. Student registration, payment records, and documentation are all maintained in physical or basic digital formats. This system causes many issues, including delays, difficulty in finding specific records, and the risk of data loss. For example, if a student's document is missing, the admin has to search through all records manually to identify and fix the issue. These manual tasks take a lot of time and make the overall process slow and stressful. To address these issues, we created the SLIITNU Panel a web-based application that digitalizes these processes and reduces the manual workload.

3.1.1 People Affected

The people most affected by the current manual system are the administrative staff and the students:

- **Administrators** must store everything in Excel sheets or notebooks. If there is a mistake or missing information, they have to check all records manually, which takes a lot of time. Additionally, they must send emails one by one to students regarding registration updates or approvals, which delays communication.

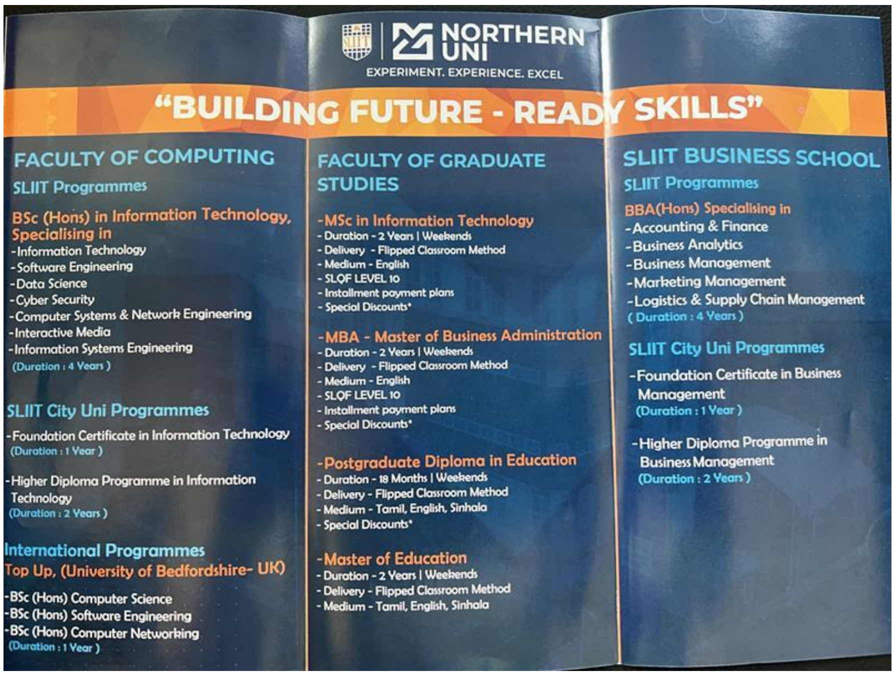
- **Students** are also affected because, even after registering through the official university website, they still need to complete manual forms inside the campus. After registration, they may have to wait a long time for confirmation since the admin sends emails manually. This delay can cause students to feel that their registration form was not received or processed. Furthermore, students do not get a clear view of the full registration process.

3.1.2 Limitations of Current Manual System

- All student data, payments, and documents are recorded in notebooks or Excel sheets.
- If a document is lost, the admin must manually go through all files to find and fix the issue.
- Admins need to send confirmation or registration approval emails manually to each student.
- Communication delays occur because emails are not automated.
- Students cannot fully track their registration status.
- Admin workload is high due to repetitive tasks.
- There is a high chance of human error and data misplacement.

These limitations highlight the need for a more efficient, secure, and streamlined system.

Below are a few examples of the current manual processes:



NORTHERN UNI
EXPERIMENT. EXPERIENCE. EXCEL

"BUILDING FUTURE - READY SKILLS"

FACULTY OF COMPUTING

SLIIT Programmes

BSc (Hons) in Information Technology, Specialising in
- Information Technology
- Software Engineering
- Data Science
- Cyber Security
- Computer Systems & Network Engineering
- Interactive Media
- Information Systems Engineering

(Duration : 4 Years)

SLIIT City Uni Programmes

- Foundation Certificate in Information Technology
(Duration : 1 Year)

- Higher Diploma Programme in Information Technology
(Duration : 2 Years)

International Programmes
Top Up, (University of Bedfordshire- UK)

- BSc (Hons) Computer Science
- BSc (Hons) Software Engineering
- BSc (Hons) Computer Networking

(Duration : 1 Year)

FACULTY OF GRADUATE STUDIES

- MSc in Information Technology
  - Duration - 2 Years | Weekends
  - Delivery - Flipped Classroom Method
  - Medium - English
  - SLQF LEVEL 10
  - Installment payment plans
  - Special Discounts*
- MBA - Master of Business Administration
  - Duration - 2 Years | Weekends
  - Delivery - Flipped Classroom Method
  - Medium - English
  - SLQF LEVEL 10
  - Installment payment plans
  - Special Discounts*
- Postgraduate Diploma in Education
  - Duration - 18 Months | Weekends
  - Delivery - Flipped Classroom Method
  - Medium - Tamil, English, Sinhala
  - Special Discounts*
- Master of Education
  - Duration - 2 Years | Weekends
  - Delivery - Flipped Classroom Method
  - Medium - Tamil, English, Sinhala

SLIIT BUSINESS SCHOOL

SLIIT Programmes

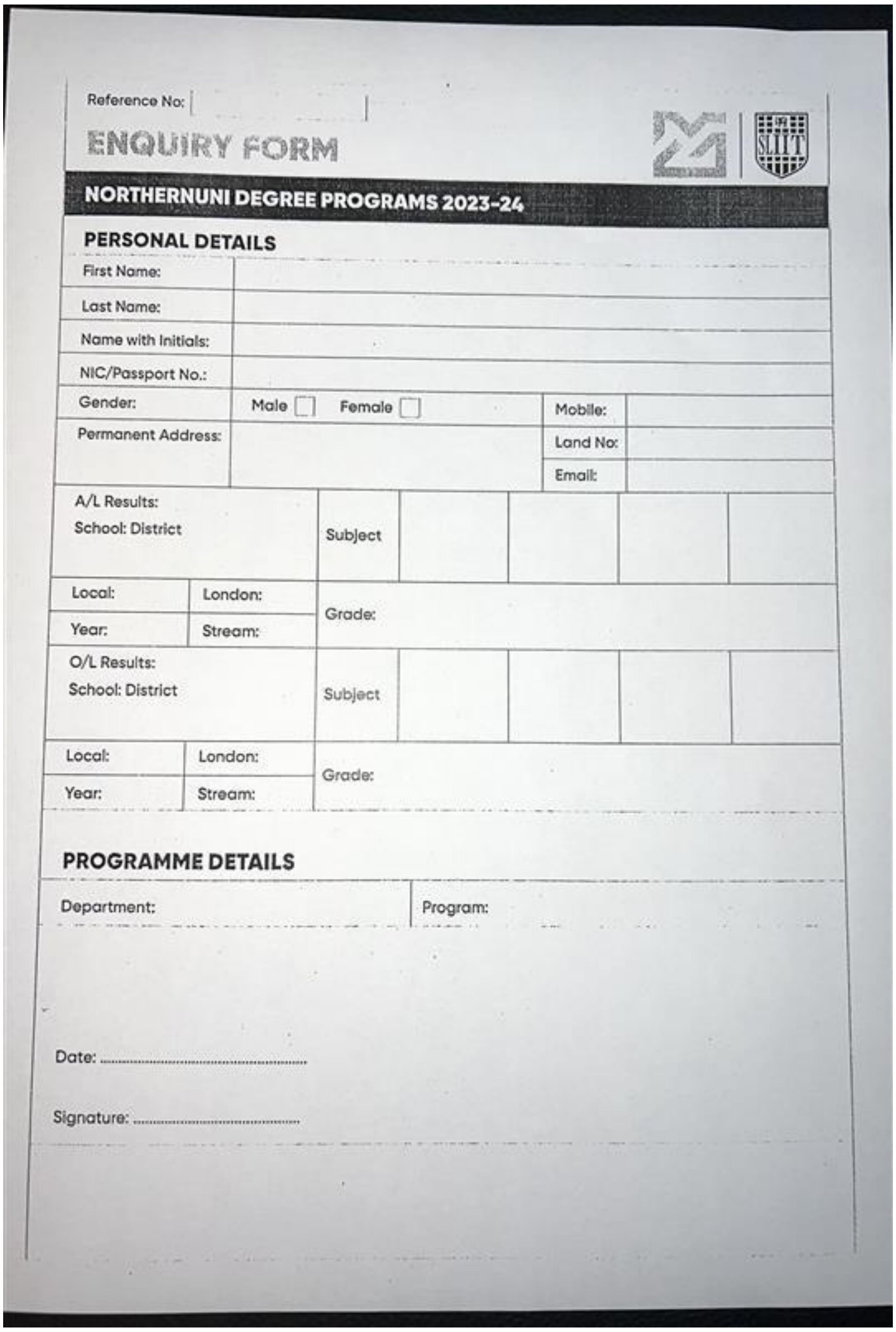
BBA(Hons) Specialising in
- Accounting & Finance
- Business Analytics
- Business Management
- Marketing Management
- Logistics & Supply Chain Management

(Duration : 4 Years)

SLIIT City Uni Programmes

- Foundation Certificate in Business Management
(Duration : 1 Year)

- Higher Diploma Programme in Business Management
(Duration : 2 Years)



Reference No:

ENQUIRY FORM

SLIIT

NORTHERNUNI DEGREE PROGRAMS 2023-24

PERSONAL DETAILS

First Name:			
Last Name:			
Name with Initials:			
NIC/Passport No.:			
Gender:	Male ☐ Female ☐	Mobile:	
Permanent Address:		Land No:	
		Email:	

A/L Results: School: District		Subject				
Local:	London:	Grade:				
Year:	Stream:					
O/L Results: School: District		Subject				
Local:	London:	Grade:				
Year:	Stream:					

PROGRAMME DETAILS

Department:	Program:

Date: ....

Signature: ....

3.2 Requirement Analysis

3.2.1 Gathering Requirements

Before starting the system design, we focused on understanding the actual needs of the users. To gather requirements effectively, we followed three main methods: arranging direct discussions with the administration staff (similar to interviews), reviewing the existing documents they use for manual operations, and conducting a detailed questionnaire session. These approaches helped us collect accurate and complete information to design a system that matches their current workflow and solves their practical problems.

3.2.1.1 Interviews

We began the requirement gathering process by directly contacting the administration staff and arranging a face-to-face meeting. During the meeting, we had an open discussion where they explained how the current student registration and record maintenance system works. Instead of a formal interview format, we interacted with them informally, asking questions and clarifying doubts as they guided us through their workflow. They shared how they manage student data manually and highlighted the difficulties they face. This interaction helped us understand the overall process from their perspective, which gave us a solid foundation to start analyzing what features they would need in a digital system.

3.2.1.2 Existing Documents

The staff also showed us various documents they currently use to store and manage data. These included notebooks, printed forms, student registration records, and course booklets. They explained how each of these is used to keep track of students, their selected courses, personal details, and exam results. We carefully reviewed these materials to understand the patterns and data fields used, such as columns for student name, ID, district, province, A/L and O/L results, and course preferences. By studying these documents, we ensured that the digitized system we design reflects the same structure, making it easier for staff to transition to the new platform without changing their workflow too much.

3.2.1.3 Questionnaire

In addition to reviewing documents and holding discussions, we conducted a detailed questionnaire session to explore further requirements. We asked staff about the additional features they wished to include beyond the current manual process. Some of their suggestions included adding dropdowns for districts and provinces to avoid manual writing, allowing students to upload payment slips instead of sending them via WhatsApp, and enabling students to download their information in PDF or Excel formats. We also asked about post-registration steps, such as email notifications, document submissions, and aptitude tests. Additionally, they requested a feature for batch-wise announcements and a notification system using Telegram for both student and staff groups. All of this feedback was documented and directly used in designing the web application to fit their practical needs.

3.2.2 General Objectives

- Provide a user-friendly system

  By digitizing the existing manual processes with the same format and structure, we ensured that users can easily use the new system without confusion.

- More reliable and more Accuracy

  Since the system eliminates manual entries and handwriting errors, it increases data accuracy and ensures that records are properly stored and retrieved.

- Ease the workload

  Automation of form filling, dropdowns, payment slip uploads, and digital document submissions has reduced the manual workload for staff.

- Speed up the work process

  Features like instant registration confirmation, document uploads, and automated email notifications help in speeding up the overall student registration process.

- Increase efficiency

  Staff can now manage student records, course selections, and announcements more effectively without switching between physical forms or WhatsApp groups.

3.2.3 Specified Objectives

- ✓ After each step of the registration process, an automated email is sent to the student to keep them informed about their status. This helps students track their process easily and reduces the number of inquiries to the administration.

- ✓ A chatbot was integrated into the web application so students can ask questions directly through the platform without needing to contact staff every time.

- ✓ The entire process of manual data entry was transformed into digital forms with structured fields that match the original manual format. This makes it familiar and easy for staff to use.

- ✓ Students are provided with a login to their dashboard where they can:

  - Able to raise batch tickets or personal tickets.
  - View their payment slips any time after uploading them.
  - Submit necessary documents.
  - See announcements and news specific to their batch or year.

- ✓ For the administration, Telegram notifications are sent instantly when a new student registers. Announcements can also be posted via the application and seen by students on their dashboards as well as via email.

- ✓ The system allows data to be exported in PDF or Excel formats, as requested by the staff, to support reporting and record-keeping needs.

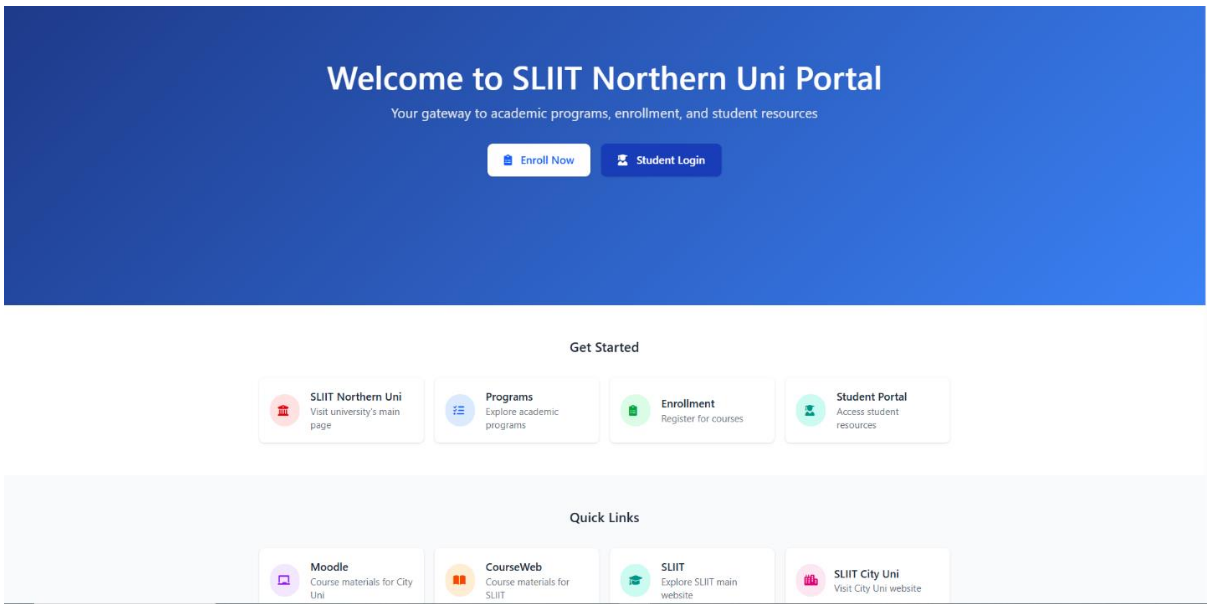
All of these were implemented based on the detailed feedback and documents provided during our requirement-gathering sessions, ensuring that the system perfectly fits their needs while improving the overall process.

3.3 Results

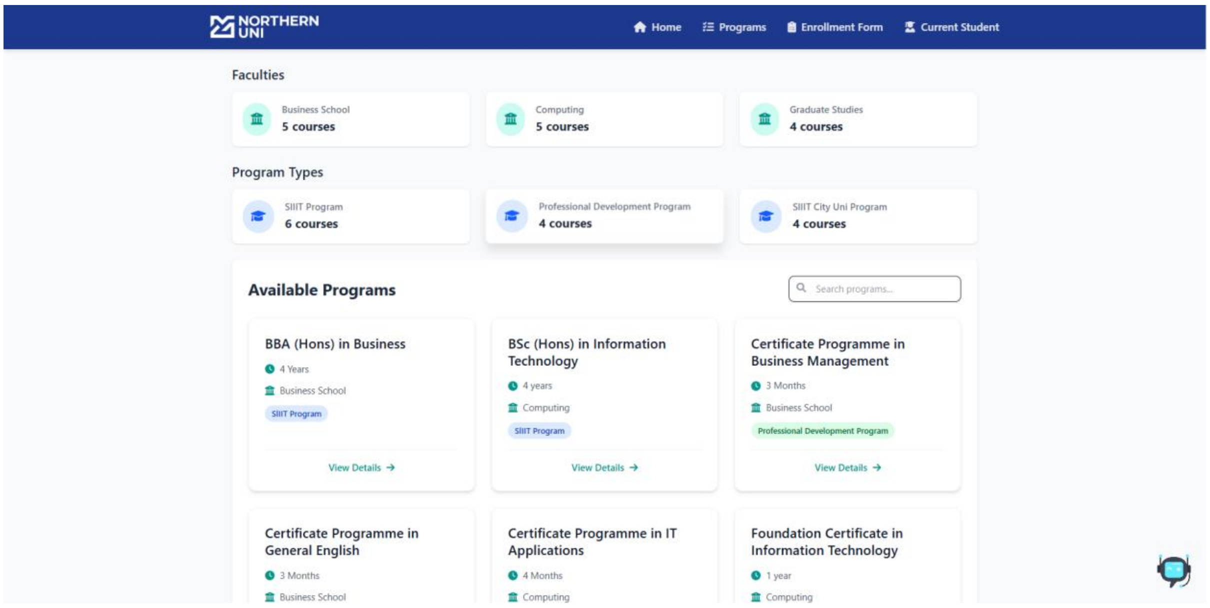
As of now, we have successfully implemented the key features of the web application based on the requirements and objectives discussed earlier. Below are some important screenshots showcasing the functionality and design of the application. These images represent the major components of the system that have been developed up to this point, demonstrating how the application meets the requirements and objectives outlined during the project.

Web Application Overview

Home Page


Welcome to SLIIT Northern Uni Portal
Your gateway to academic programs, enrollment, and student resources
Enroll Now
Student Login
Get Started
SLIIT Northern Uni
Visit university's main page
Programs
Explore academic programs
Enrollment
Register for courses
Student Portal
Access student resources
Quick Links
Moodle
Course materials for City Uni
CourseWeb
Course materials for SLIIT
SLIIT
Explore SLIIT main website
SLIIT City Uni
Visit City Uni website

Program Page



Faculties

Business School 5 courses | Computing 5 courses | Graduate Studies 4 courses

Program Types

SIIT Program 6 courses | Professional Development Program 4 courses | SIIT City Uni Program 4 courses

Available Programs

BBA (Hons) in Business
4 Years
Business School
SIIT Program
View Details →

BSc (Hons) in Information Technology
4 years
Computing
SIIT Program
View Details →

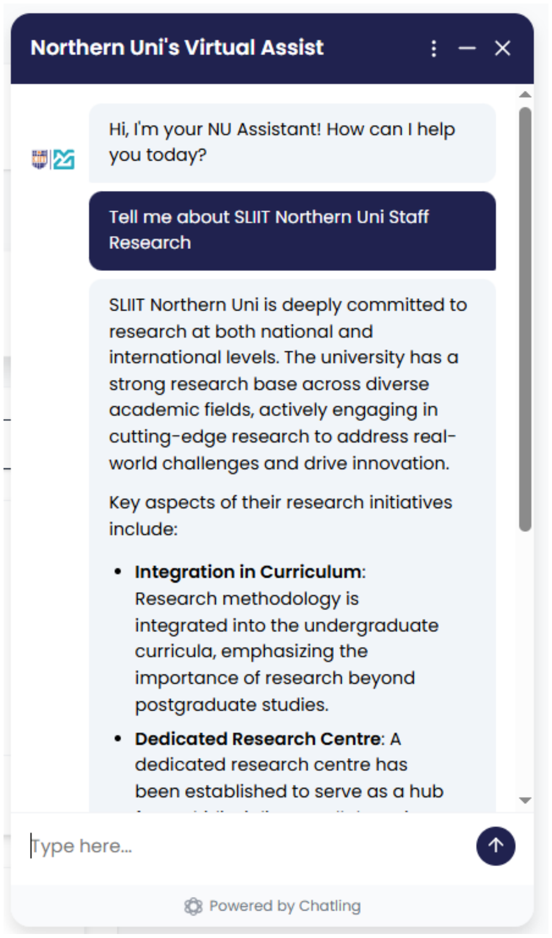
Certificate Programme in Business Management
3 Months
Business School
Professional Development Program
View Details →

Certificate Programme in General English
3 Months
Business School

Certificate Programme in IT Applications
4 Months
Computing

Foundation Certificate in Information Technology
1 year
Computing

Chatbot (Northern Uni's Virtual Assist)



Hi, I'm your NU Assistant! How can I help you today?

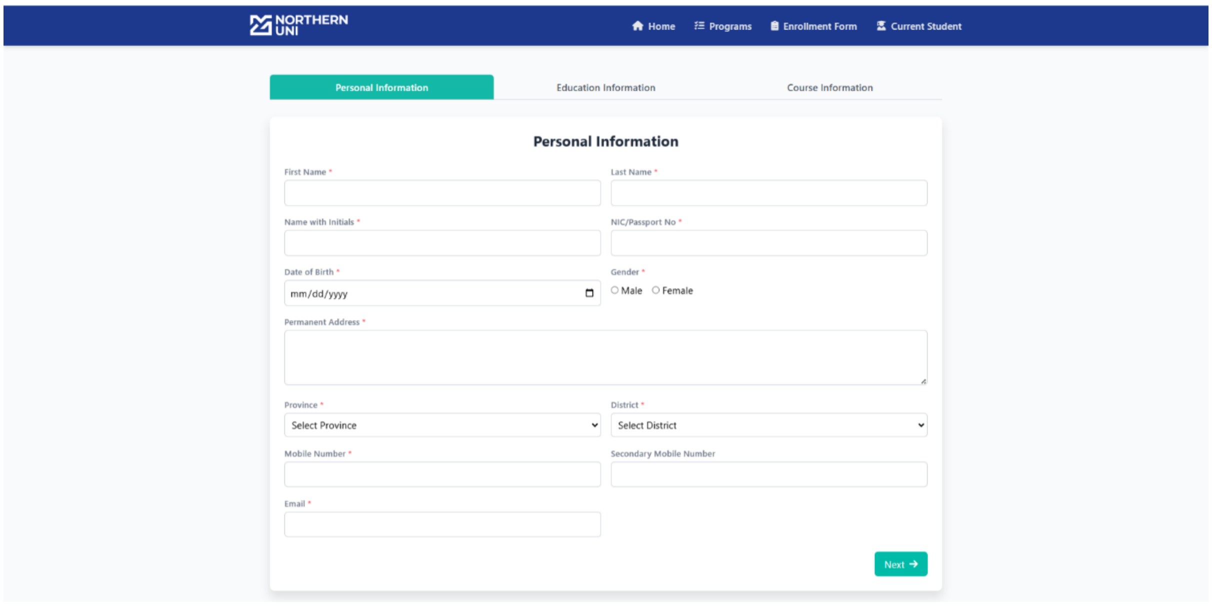
Tell me about SLIIT Northern Uni Staff Research

SLIIT Northern Uni is deeply committed to research at both national and international levels. The university has a strong research base across diverse academic fields, actively engaging in cutting-edge research to address real-world challenges and drive innovation.

Key aspects of their research initiatives include:

- **Integration in Curriculum**: Research methodology is integrated into the undergraduate curricula, emphasizing the importance of research beyond postgraduate studies.
- **Dedicated Research Centre**: A dedicated research centre has been established to serve as a hub

Enrollment Form Page


NORTHERN UNI

Home · Programs · Enrollment Form · Current Student

Personal Information · Education Information · Course Information

Personal Information

First Name *

Last Name *

Name with Initials *

NIC/Passport No *

Date of Birth *

mm/dd/yyyy

Gender *

○ Male ○ Female

Permanent Address *

Province *

Select Province

District *

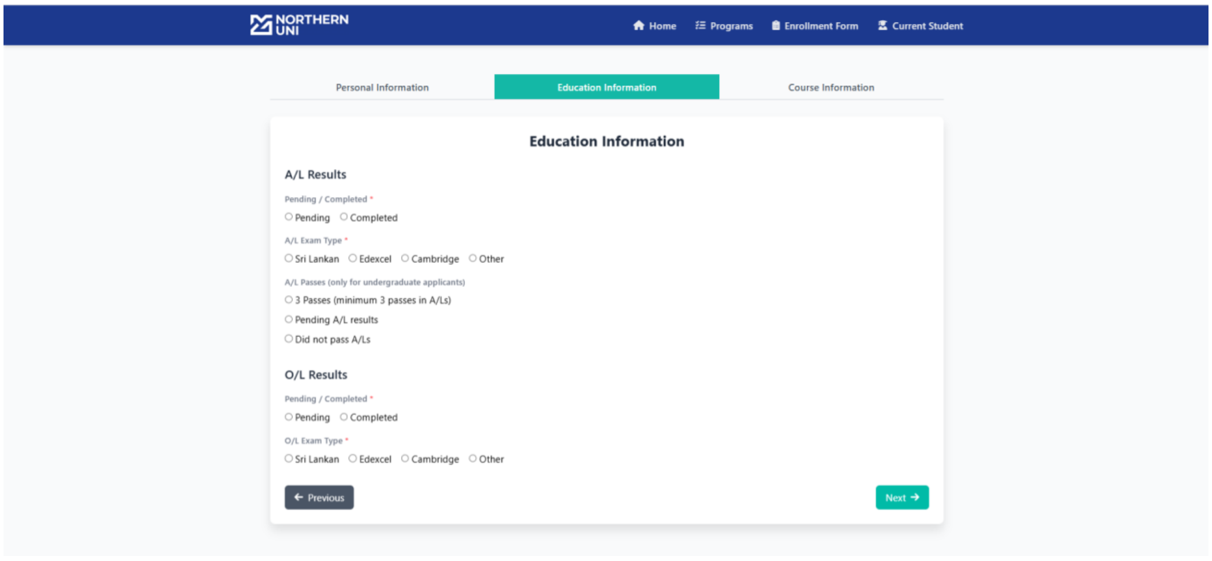
Select District

Mobile Number *

Secondary Mobile Number

Email *

Next →


NORTHERN UNI

Home · Programs · Enrollment Form · Current Student

Personal Information · Education Information · Course Information

Education Information

A/L Results

Pending / Completed *

○ Pending ○ Completed

A/L Exam Type *

○ Sri Lankan ○ Edexcel ○ Cambridge ○ Other

A/L Passes (only for undergraduate applicants)

○ 3 Passes (minimum 3 passes in A/Ls)

○ Pending A/L results

○ Did not pass A/Ls

O/L Results

Pending / Completed *

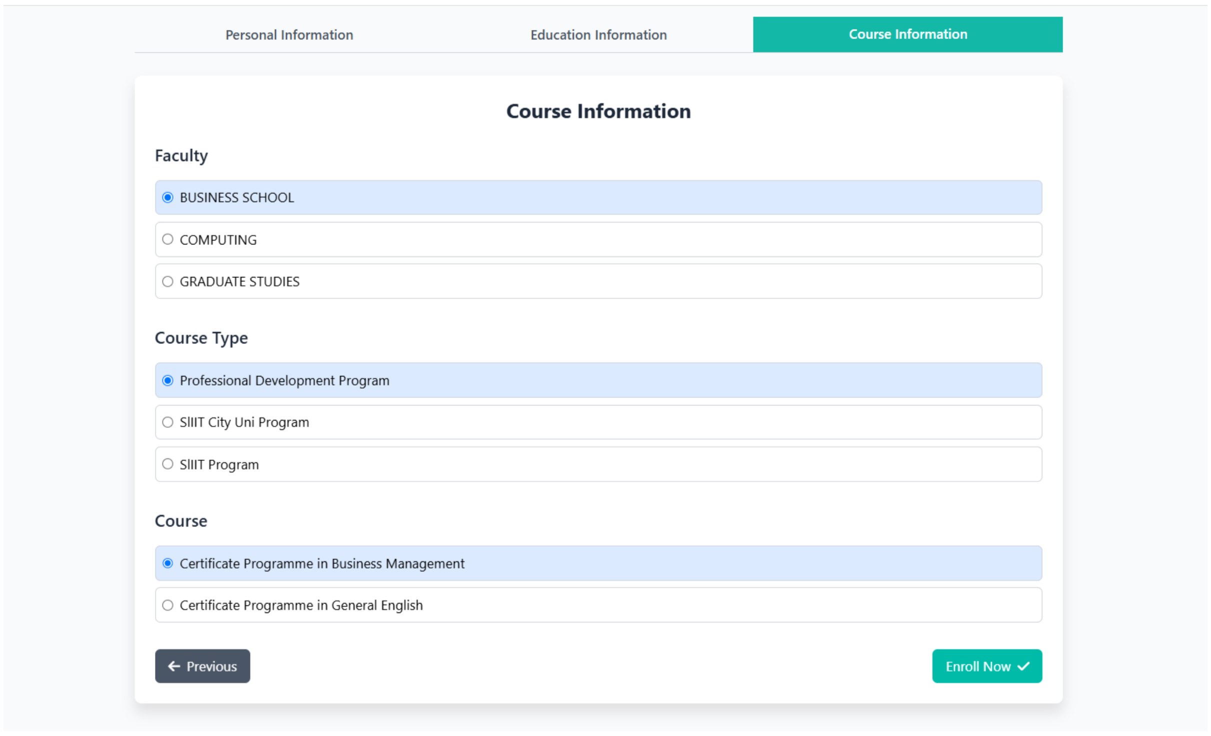
○ Pending ○ Completed

O/L Exam Type *

○ Sri Lankan ○ Edexcel ○ Cambridge ○ Other

← Previous

Next →



Personal Information | Education Information | Course Information

Course Information

Faculty

- [✓ BUSINESS SCHOOL]
- COMPUTING
- GRADUATE STUDIES

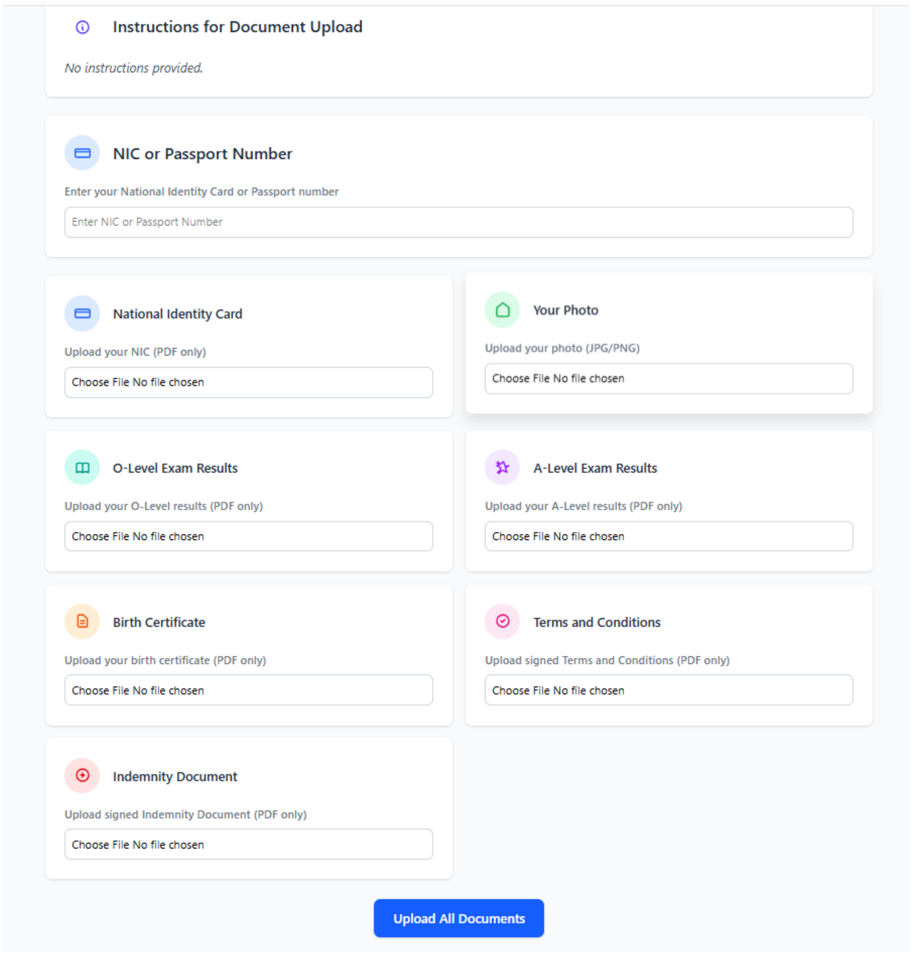
Course Type

- [✓ Professional Development Program]
- SIIIT City Uni Program
- SIIIT Program

Course

- [✓ Certificate Programme in Business Management]
- Certificate Programme in General English

Upload Document Page



Instructions for Document Upload

No instructions provided.

NIC or Passport Number

Enter your National Identity Card or Passport number

Enter NIC or Passport Number

National Identity Card

Upload your NIC (PDF only)

Choose File No file chosen

Your Photo

Upload your photo (JPG/PNG)

Choose File No file chosen

O-Level Exam Results

Upload your O-Level results (PDF only)

Choose File No file chosen

A-Level Exam Results

Upload your A-Level results (PDF only)

Choose File No file chosen

Birth Certificate

Upload your birth certificate (PDF only)

Choose File No file chosen

Terms and Conditions

Upload signed Terms and Conditions (PDF only)

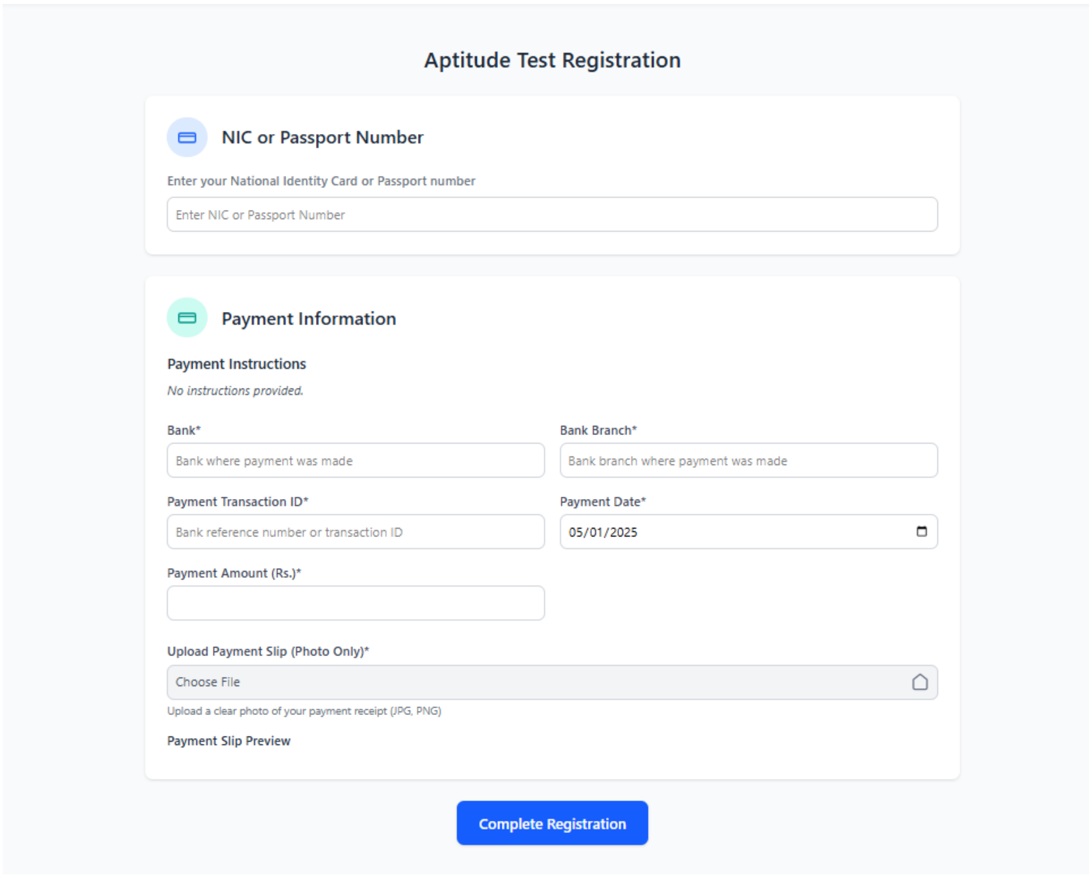
Choose File No file chosen

Indemnity Document

Upload signed Indemnity Document (PDF only)

Choose File No file chosen

Aptitude Page


Aptitude Test Registration

NIC or Passport Number

Enter your National Identity Card or Passport number

Enter NIC or Passport Number

Payment Information

Payment Instructions

No instructions provided.

Bank*

Bank where payment was made

Bank Branch*

Bank branch where payment was made

Payment Transaction ID*

Bank reference number or transaction ID

Payment Date*

05/01/2025

Payment Amount (Rs.)*

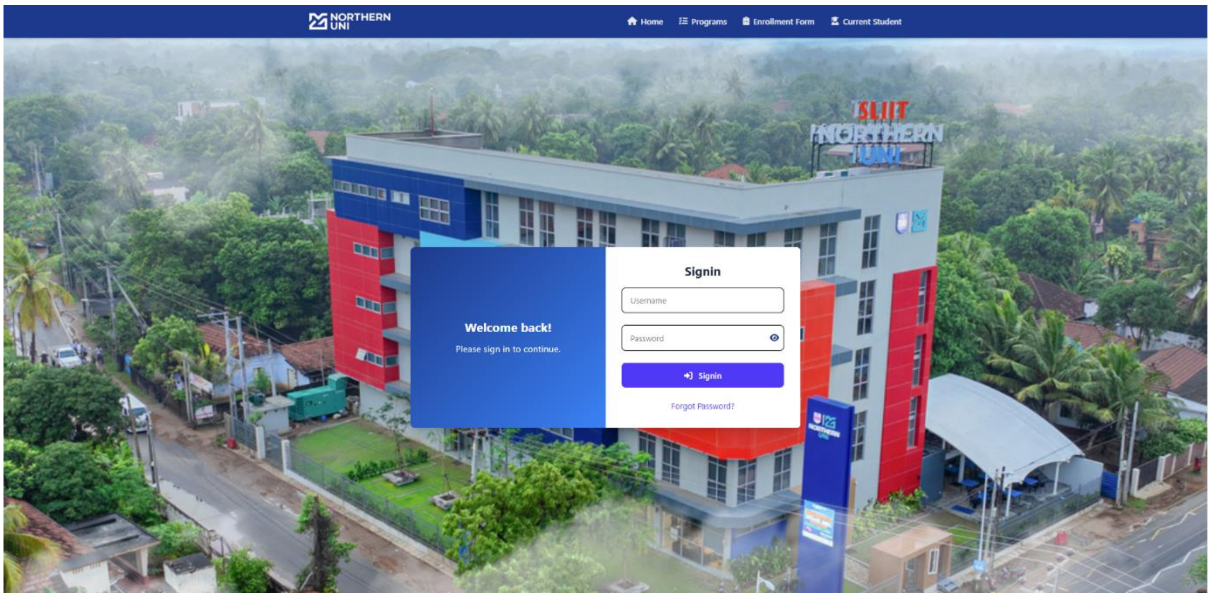
Upload Payment Slip (Photo Only)*

Choose File

Upload a clear photo of your payment receipt (JPG, PNG)

Payment Slip Preview

Current Student Login Page

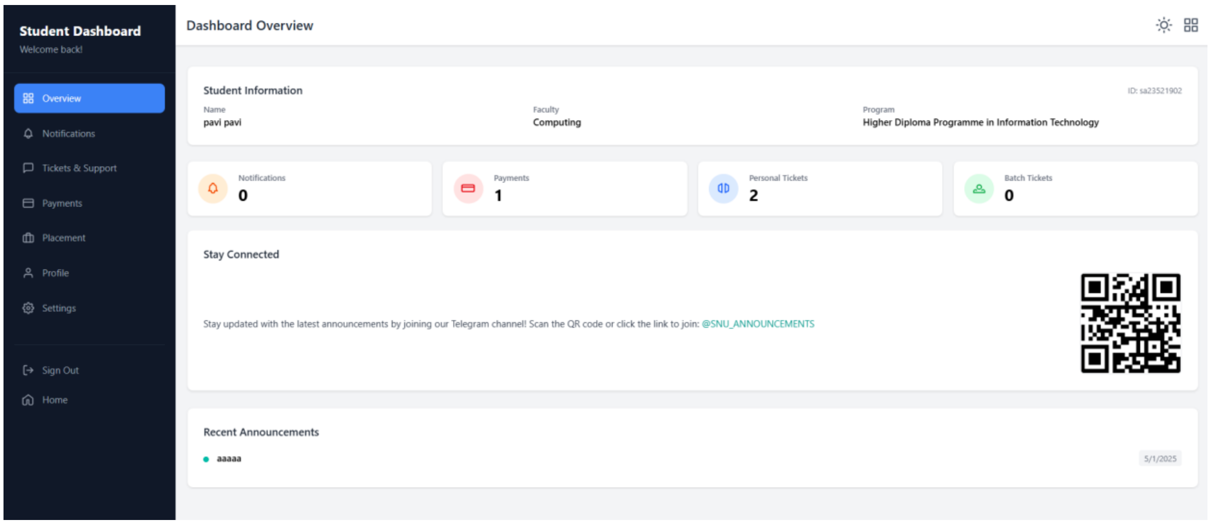


Welcome back!

Please sign in to continue.

Signin

Student Dashboard View



Dashboard Overview

Student Information

ID: sa23521902

Name: pavi pavi
Faculty: Computing
Program: Higher Diploma Programme in Information Technology

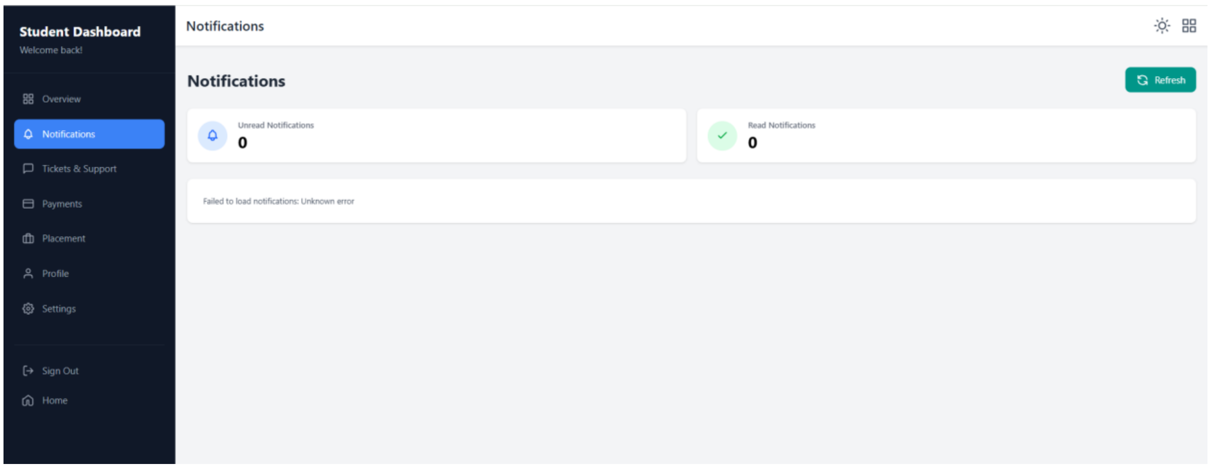
Notifications 0
Payments 1
Personal Tickets 2
Batch Tickets 0

Stay Connected

Stay updated with the latest announcements by joining our Telegram channel! Scan the QR code or click the link to join: @SNU_ANNOUNCEMENTS

Recent Announcements

● aaaaa 5/1/2025

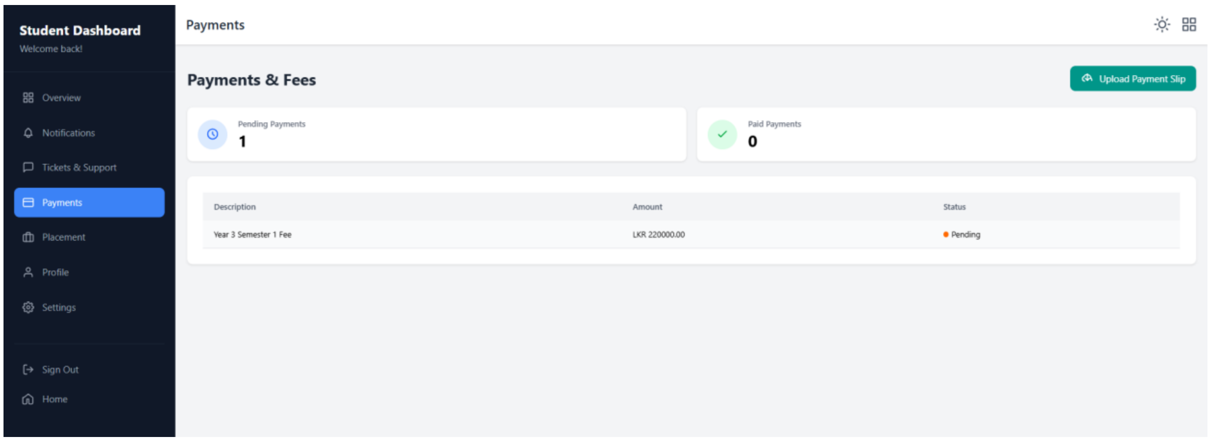


Notifications

Notifications

Unread Notifications 0
Read Notifications 0

Failed to load notifications: Unknown error

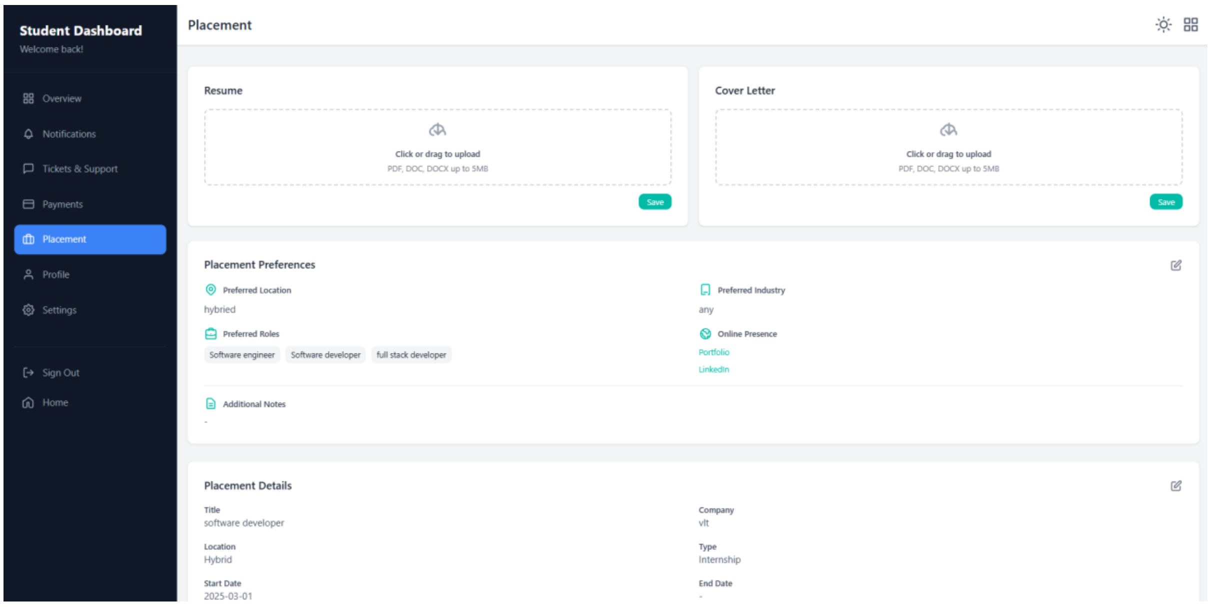


Payments

Payments & Fees

Pending Payments 1
Paid Payments 0

Description	Amount	Status
Year 3 Semester 1 Fee	LKR 220000.00	● Pending



Placement

Resume

Click or drag to upload
PDF, DOC, DOCX up to 5MB
Save

Cover Letter

Click or drag to upload
PDF, DOC, DOCX up to 5MB
Save

Placement Preferences

Preferred Location
hybried

Preferred Industry
any

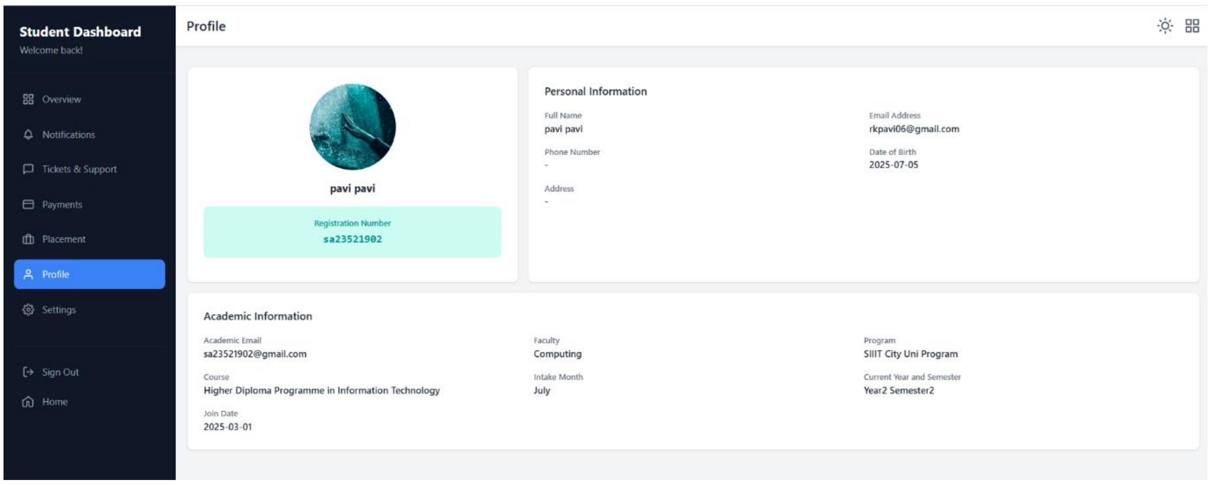
Preferred Roles
Software engineer | Software developer | full stack developer

Online Presence
Portfolio
LinkedIn

Additional Notes
-

Placement Details

Title software developer	Company vlt
Location Hybrid	Type Internship
Start Date 2025-03-01	End Date -



Profile

pavi pavi

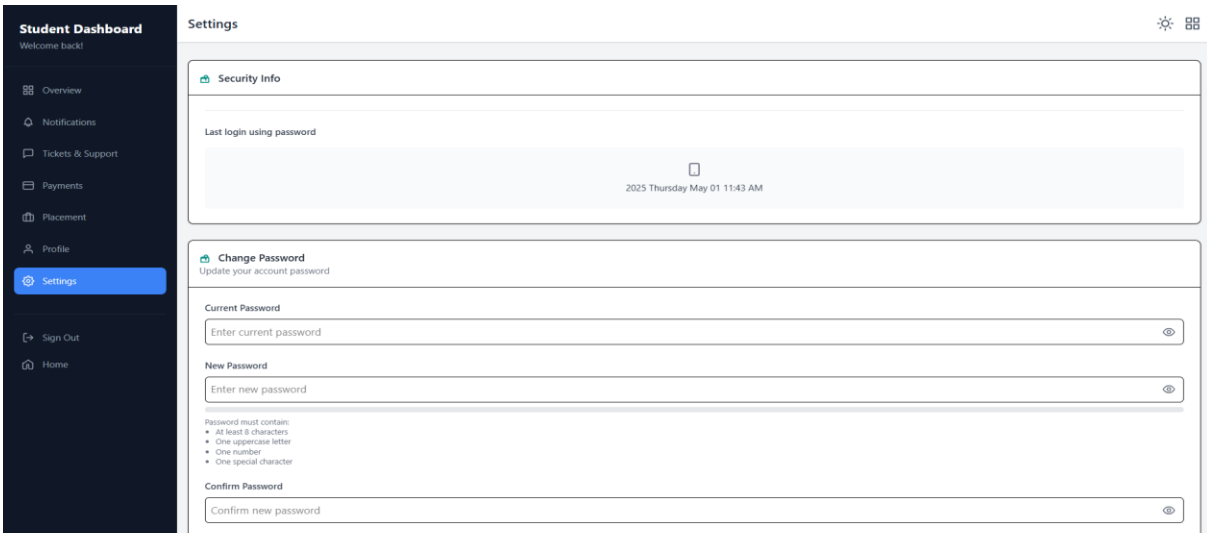
Registration Number
sa23521902

Personal Information

Full Name pavi pavi	Email Address rkpavi06@gmail.com
Phone Number -	Date of Birth 2025-07-05
Address -	

Academic Information

Academic Email sa23521902@gmail.com	Faculty Computing	Program SIIIT City Uni Program
Course Higher Diploma Programme in Information Technology	Intake Month July	Current Year and Semester Year2 Semester2
Join Date 2025-03-01		



Settings

Security Info

Last login using password

2025 Thursday May 01 11:43 AM

Change Password
Update your account password

Current Password
Enter current password

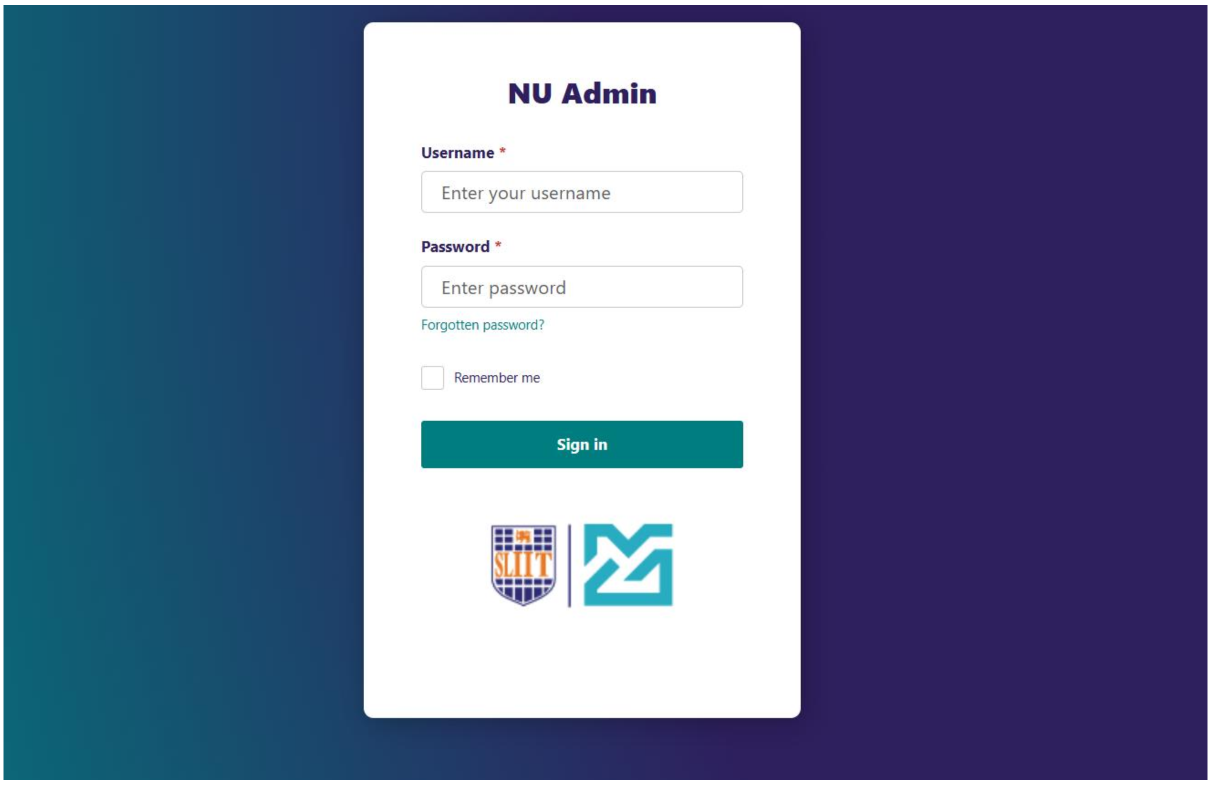
New Password
Enter new password

Password must contain:
- At least 8 characters
- One uppercase letter
- One number
- One special character

Confirm Password
Confirm new password

Administration View

Admin Login Page



NU Admin

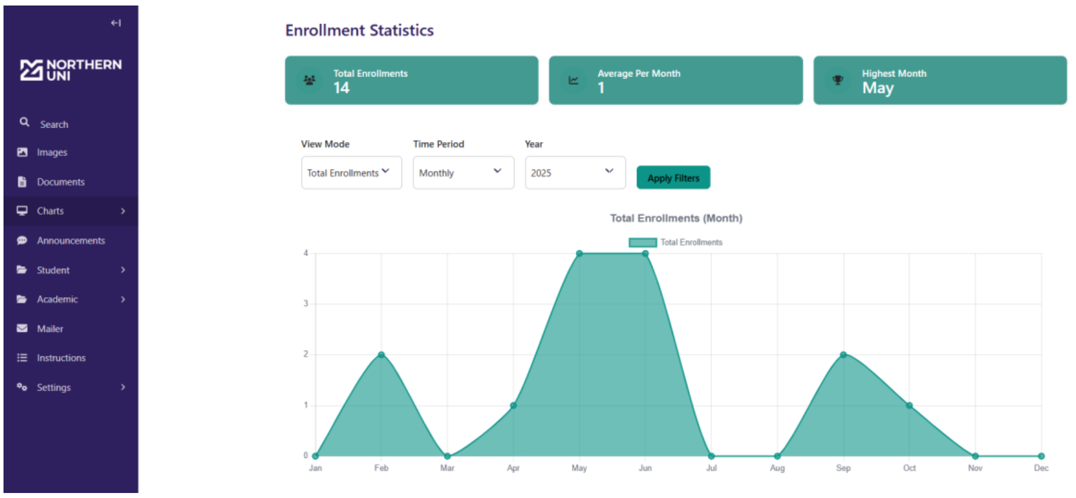
Username *

Enter your username

Password *

Enter password

Admin Charts



Enrollment Statistics

Total Enrollments 14

Average Per Month 1

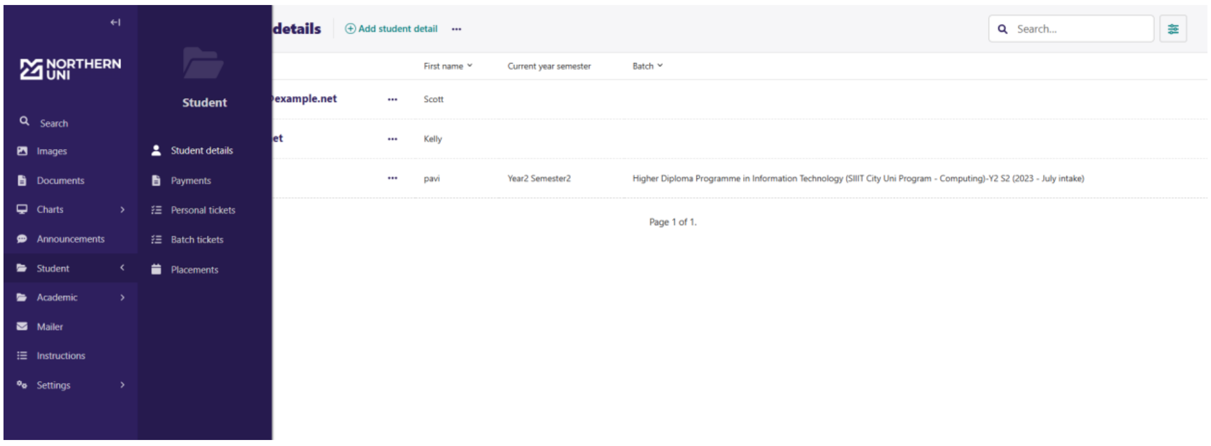
Highest Month May

View Mode: Total Enrollments

Time Period: Monthly

Year: 2025

Student Page

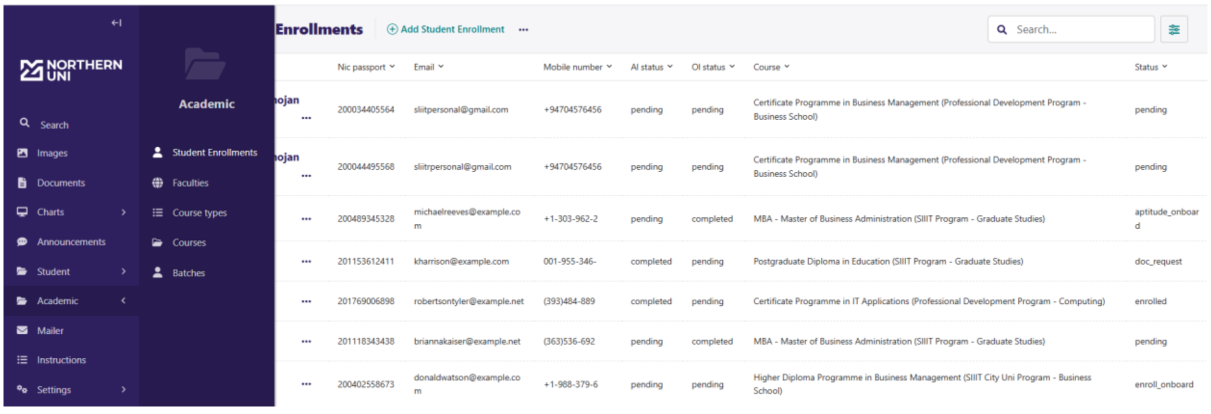


details

		First name	Current year semester	Batch
example.net	•••	Scott		
et	•••	Kelly		
	•••	pavi	Year2 Semester2	Higher Diploma Programme in Information Technology (SIIIT City Uni Program - Computing)-Y2 S2 (2023 - July intake)

Page 1 of 1.

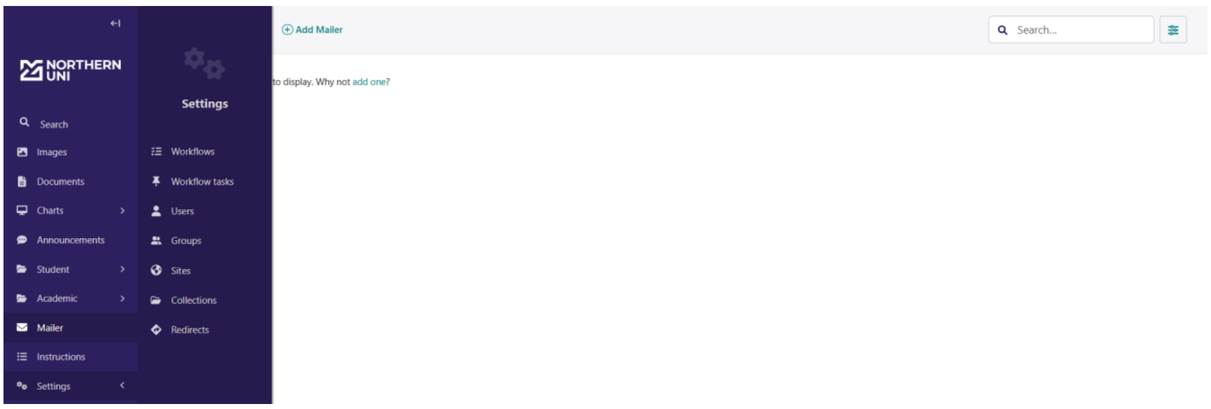
Academic Page



Enrollments

	Nic passport	Email	Mobile number	Al status	Ol status	Course	Status
ojan •••	200034405564	sliitpersonal@gmail.com	+94704576456	pending	pending	Certificate Programme in Business Management (Professional Development Program - Business School)	pending
ojan •••	200044495568	sliitrpersonal@gmail.com	+94704576456	pending	pending	Certificate Programme in Business Management (Professional Development Program - Business School)	pending
•••	200489345328	michaelreeves@example.com	+1-303-962-2	pending	completed	MBA - Master of Business Administration (SIIIT Program - Graduate Studies)	aptitude_onboard
•••	201153612411	kharrison@example.com	001-955-346-	completed	pending	Postgraduate Diploma in Education (SIIIT Program - Graduate Studies)	doc_request
•••	201769006898	robertsontyler@example.net	(393)484-889	completed	pending	Certificate Programme in IT Applications (Professional Development Program - Computing)	enrolled
•••	201118343438	briannakaiser@example.net	(363)536-692	pending	completed	MBA - Master of Business Administration (SIIIT Program - Graduate Studies)	pending
•••	200402558673	donaldwatson@example.com	+1-988-379-6	pending	pending	Higher Diploma Programme in Business Management (SIIIT City Uni Program - Business School)	enroll_onboard

Settings Page



to display. Why not add one?

4 Design and Implementation

After gathering the requirements and developing the application step-by-step, the design and implementation phase helped us bring the final solution to life. Based on the needs collected from the administration and students, we designed the system to transform manual tasks into fully digitized processes. We carefully mapped out each module from registration to document submission, payment handling, and announcements ensuring each feature was implemented in a user-friendly and efficient way.

In this stage, we mainly focused on translating the system requirements into actual components using proper design strategies. Each interface and function was built to mirror the original manual system’s structure so users could quickly adapt. The student dashboard, admin panel, and form handling were designed to reduce staff workload and improve process speed. Important features such as automated emails, PDF/Excel exports, Telegram notifications, and chatbots were implemented during this phase.

4.1 High Level Architecture Design

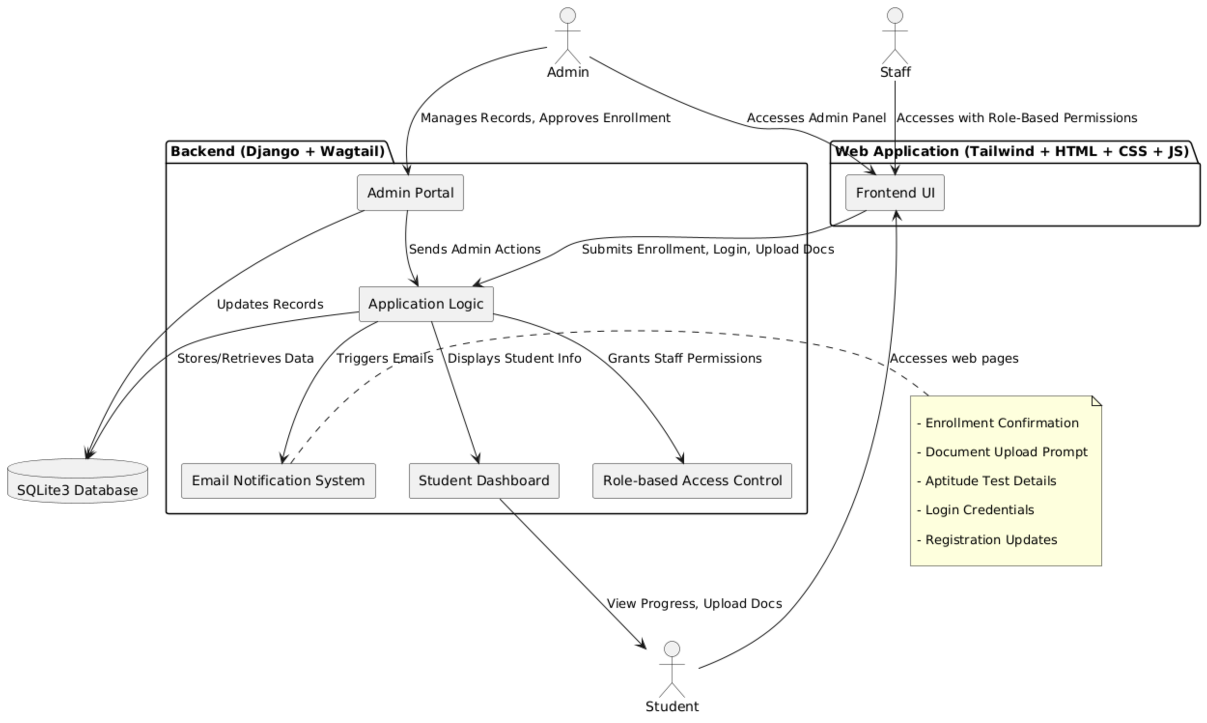


Figure 4.1: High Level Architecture Diagram of the System

4.2 Hardware, Software and Communication Interfaces

4.2.1 Hardware Requirements

To build and test this system, we mainly used personal laptops, desktop PCs, and mobile phones. During the development process, we verified the responsiveness and performance of the system across different screen sizes and devices. The admin side of the system was mainly tested using PCs and laptops for better visibility and control, while the student dashboard and other user interfaces were also checked using mobile devices to ensure they work smoothly on phones.

4.2.2 Software Requirements

For the development, we used Visual Studio Code as our main code editor. The front-end was designed using HTML and Tailwind CSS, and the back-end was developed using Django and Wagtail. The system was tested and run on Windows, Linux, and macOS environments since our team used different operating systems during development. For managing the database, we used SQLite3 in the development stage.

4.2.3 Communication Interfaces

Since this is a web-based system, all communication between the client and server happens over HTTP/HTTPS protocols. Students and admins interact with the system through a browser-based interface. For internal communication, like notifying students or admins, the system uses email notifications and Telegram alerts. These were integrated to keep users informed throughout each step of the registration and document submission process.

4.3 Non-Functional Requirements

We made sure the system is user-friendly and easy to navigate by following a clean and simple layout. The goal was to ensure that even users who are not very tech-savvy can use the system without much difficulty. We also made sure the system loads fast and works across different devices and browsers. Data storage is handled securely, and backups can be taken when needed.

4.3.1 Operational Requirements

The system must be hosted on a server with internet access, and users should have a stable connection to access the application. Admin users need access to the backend interface, while students can log in through the public front-end portal. All operations like document uploads, registration, and announcements should work smoothly with real-time updates and notifications.

4.3.2 Performance Requirements

We designed the system to perform well even if many students are using it at the same time, especially during registration periods. Pages load within seconds, and backend processes like form submissions, email notifications, and file uploads are handled efficiently without delays. We also optimized the database structure to support fast searching and exporting of records.

4.3.3 Security Requirements

Security was an important part of the system. User passwords are stored in hashed format. Each user (student, staff, or admin) only has access to the features relevant to their role using role-based access control. The document upload section is protected, and only verified users can access or modify sensitive information. We also implemented input validation to avoid unwanted data entries and reduce common web security issues like form tampering.

4.4 Implementation

4.4.1 Development Methodology

For this project, we followed the Agile methodology because it helped us to handle the development process in a more flexible and efficient way. Since our team had to build both the frontend and backend while regularly testing features, Agile was the best choice for us. We didn’t do everything at once instead, we split the work into smaller parts and completed it step by step.

After gathering the requirements clearly from the client, we first worked on the registration and enrollment flow. Once that was done and tested, we moved to the document upload, student dashboard, and admin functionalities like announcements and student details. Each module was developed separately, and after completing one part, we tested it, fixed issues if needed, and then moved to the next one. Also, every time we finished a part, we showed it to our team and got feedback, which helped us to improve further.

This way, by breaking the work into smaller tasks and doing regular testing and reviewing, we were able to complete the project without confusion or last-minute issues. So, using Agile really supported us in organizing our work and making sure we met all the requirements smoothly.

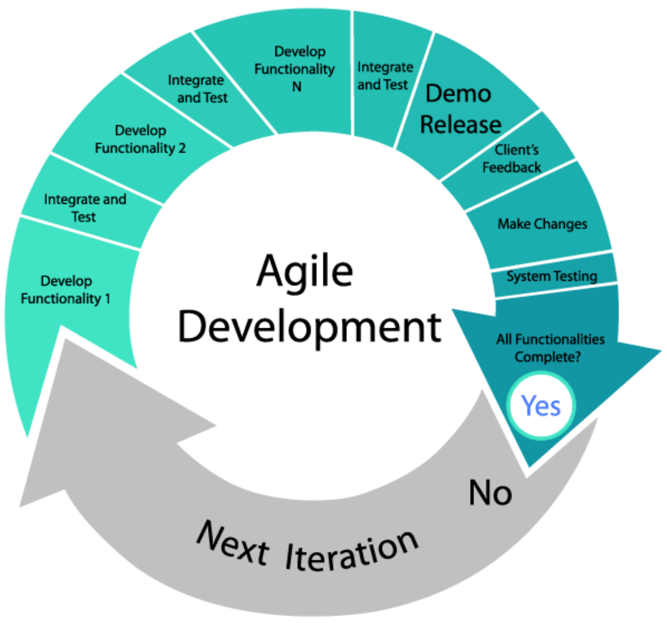


Figure 4.2: Development Methodology Agile Methodology [4]

4.4.2 Development Technologies

4.4.2.1 Coding Environment

- ✓ For this project, we mainly worked on the front-end using **HTML, CSS, JavaScript, and Tailwind CSS**, and for the back-end, we used **Django** along with **Wagtail** as the CMS framework. These technologies together helped us to build a full-stack web application that handles everything from user interfaces to backend processing. We structured and handled different parts of the code in a modular and understandable way, so it's easy to maintain and update in the future as well.

4.4.2.2 Database Environment

- ✓ For the database, we used **SQLite3** since it's lightweight and well-suited for development and testing purposes. It also integrates well with Django, and we didn't require a heavy or complex database system for our current needs. All the student details, enrollment records, and admin-side operations are stored and retrieved using SQLite3.

4.4.2.3 Report Platform

- ✓ When preparing our documentation and project reports, we mainly used **Microsoft Word** and **Google Docs** to draft and collaborate. Once everything was finalized, we converted them into **PDF format** for proper submission and presentation. These tools helped us format the report neatly and make sure everything was clearly presented.

4.4.2.4 Interface Technology

- ✓ The entire front-end interface of our system was designed using **HTML, CSS**, and mainly **Tailwind CSS**. Tailwind helped us keep the design responsive and clean without spending too much time writing custom styles. The structure is straightforward and user-friendly for both students and admins.

4.4.3 Development Tools

4.4.3.1 Coding IDE

- ✓ Our coding IDE was **Visual Studio Code**, which we found really efficient and flexible for working with multiple files, testing backend logic, and running Django applications. It helped us manage everything in one place and saved us a lot of time.

4.4.3.2 Database Management System

- ✓ As mentioned before, we used **SQLite3** as the DBMS for our project. It's built-in with Django and doesn't need a separate setup, which made it perfect for our use case. It supported all the functionalities we required for storing, retrieving, and managing student and admin data.

4.4.3.3 Reporting Tools

- ✓ For reporting tools, we worked with **Microsoft Word**, **Google Docs**, and finally converted everything into **PDF** for final submissions. These platforms were helpful for writing content collaboratively, checking formatting, and ensuring the report was clean and professional.

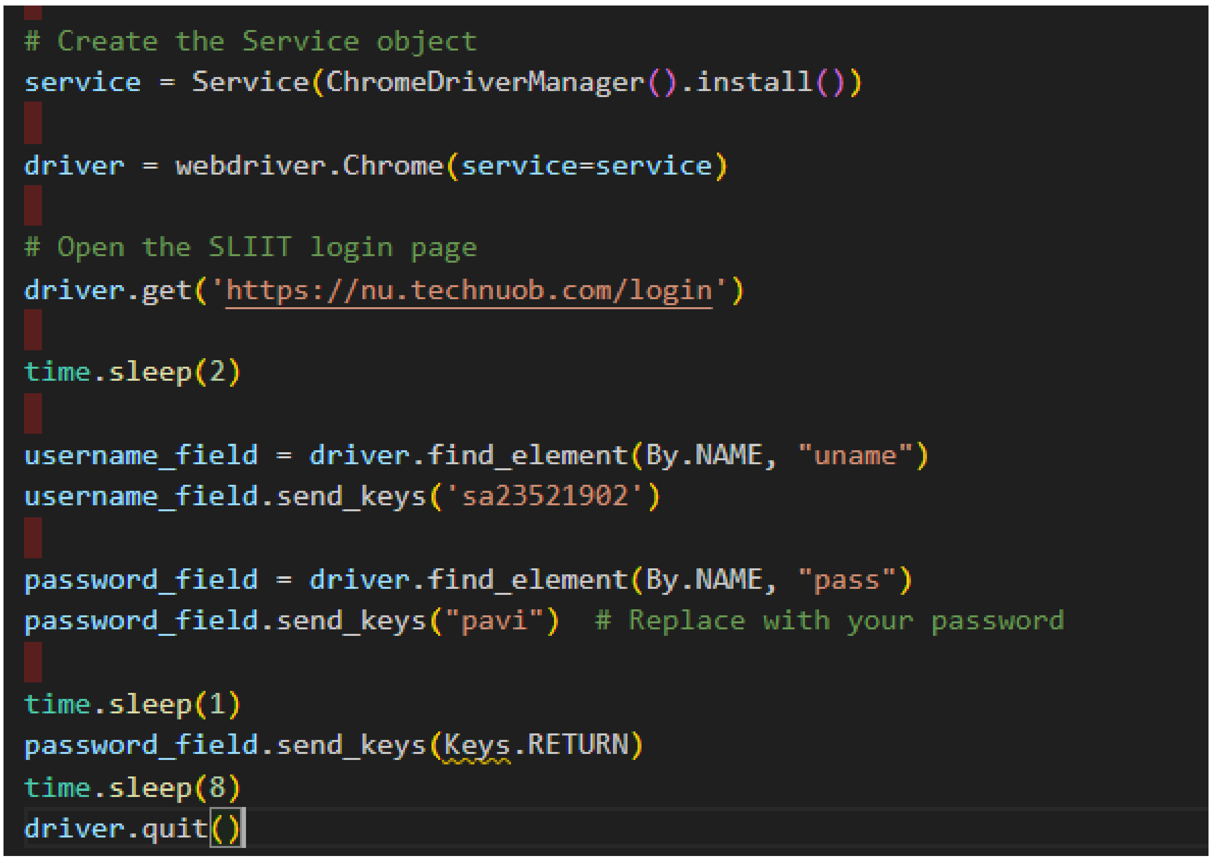
5 Testing and Deployment

We performed Selenium automated testing for all the main pages of our system, including login, enrollment, document upload, payment and dashboard features. These tests helped us check if the system works properly and responds correctly to user actions.

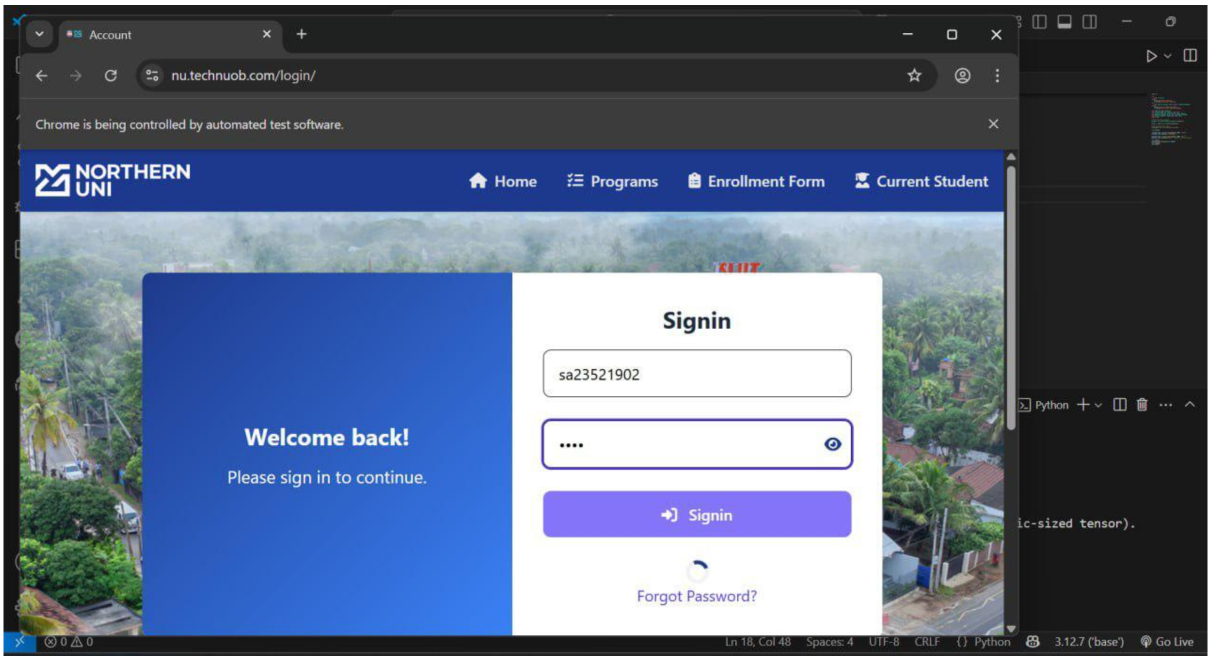
In this report, we have included sample Selenium test results for the login page and the enrollment form, to show how we verified the system functionality. These tests confirm that both pages are working correctly and are ready for real use.

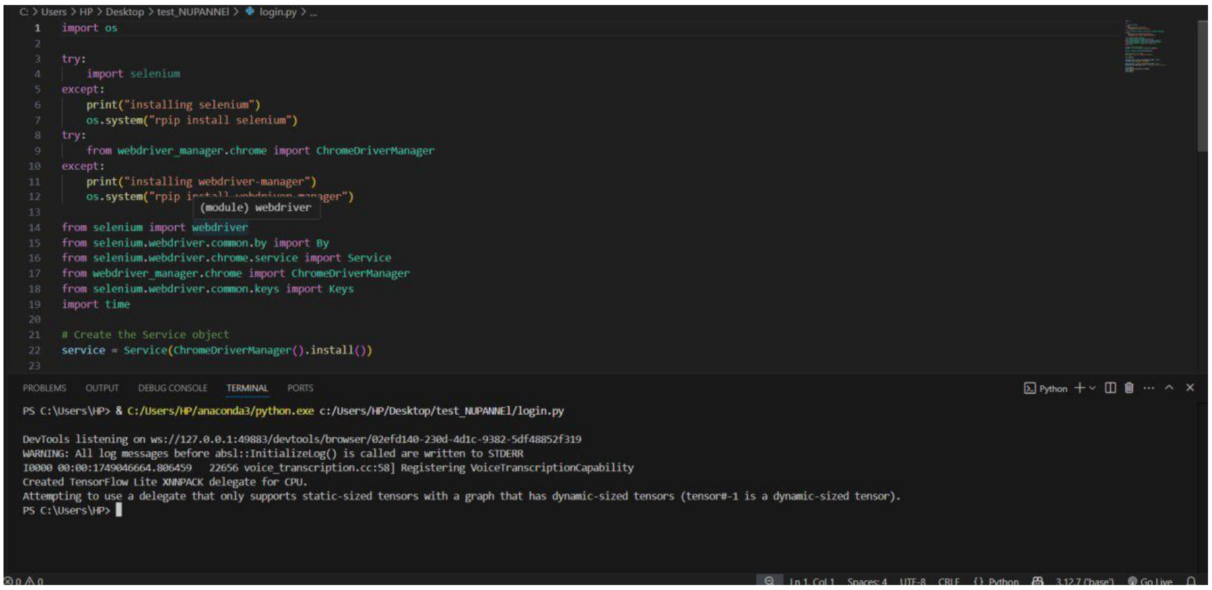
Below are the Selenium test screenshots for the login and enrollment pages:

Selenium Test for Login Page:



```
# Create the Service object
service = Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=service)

# Open the SLIIT login page
driver.get('https://nu.technuob.com/login')

time.sleep(2)

username_field = driver.find_element(By.NAME, "uname")
username_field.send_keys('sa23521902')

password_field = driver.find_element(By.NAME, "pass")
password_field.send_keys("pavi")  # Replace with your password

time.sleep(1)
password_field.send_keys(Keys.RETURN)
time.sleep(8)
driver.quit()
```

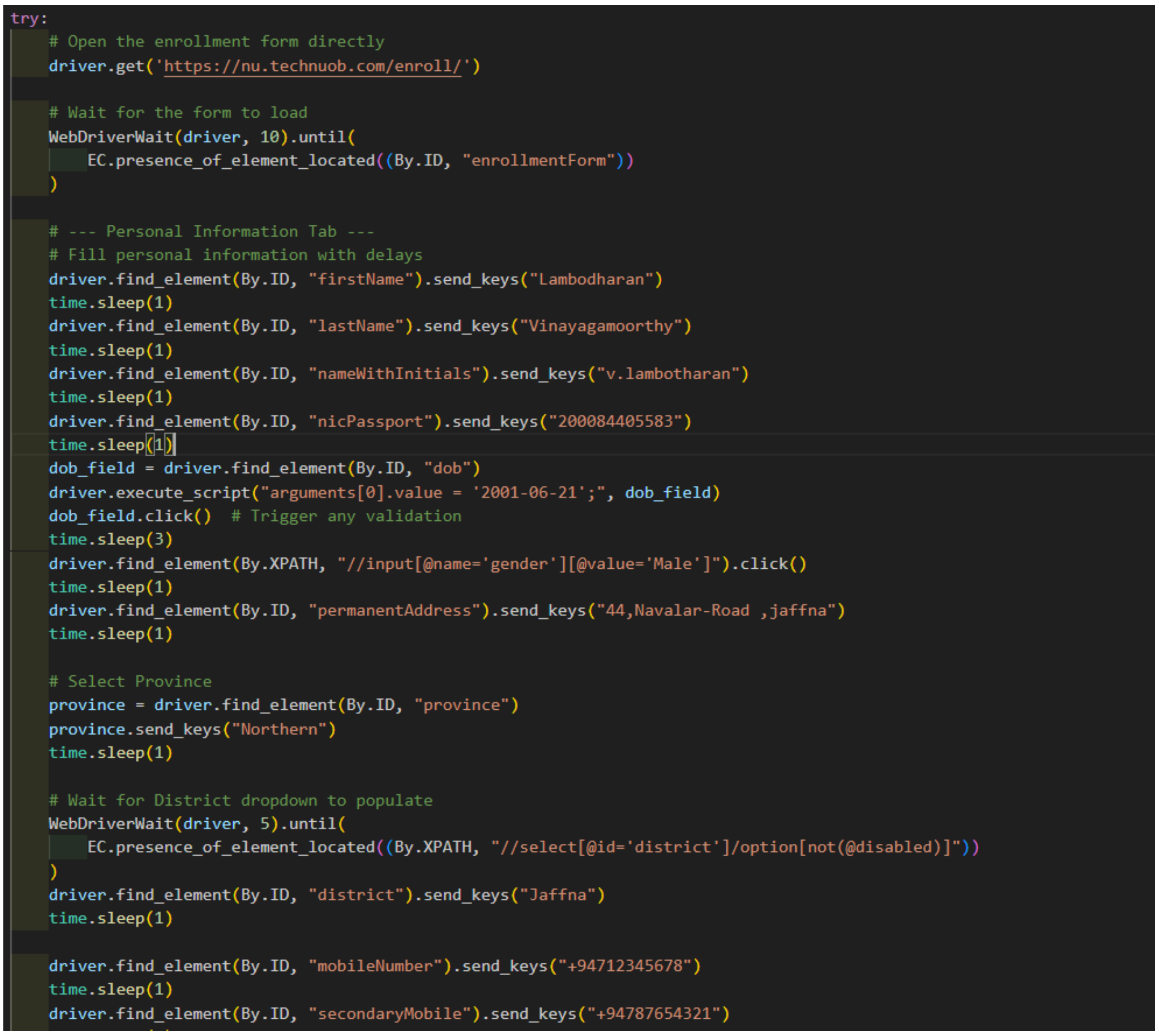



C: > Users > HP > Desktop > test_NUPANNEl > login.py > ...

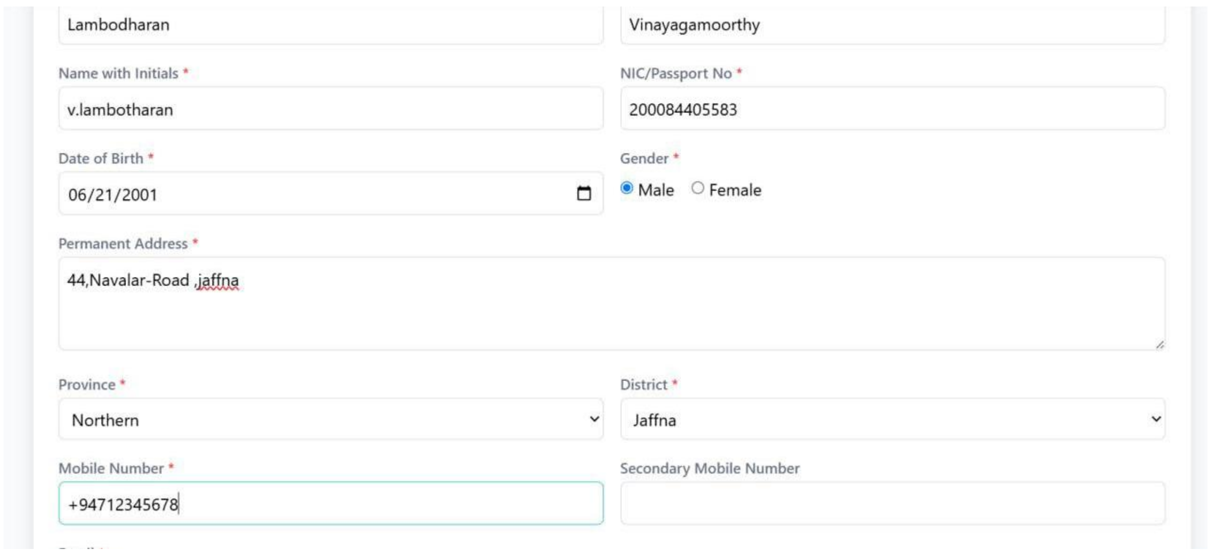
```python
import os

try:
    import selenium
except:
    print("installing selenium")
    os.system("rpip install selenium")
try:
    from webdriver_manager.chrome import ChromeDriverManager
except:
    print("installing webdriver-manager")
    os.system("rpip install webdriver-manager")

from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.keys import Keys
import time

# Create the Service object
service = Service(ChromeDriverManager().install())
```

```
PS C:\Users\HP> & C:/Users/HP/anaconda3/python.exe c:/Users/HP/Desktop/test_NUPANNEl/login.py

DevTools listening on ws://127.0.0.1:49883/devtools/browser/02efd140-230d-4d1c-9382-5df48852f319
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1749046664.806459   22656 voice_transcription.cc:58] Registering VoiceTranscriptionCapability
Created TensorFlow Lite XNNPACK delegate for CPU.
Attempting to use a delegate that only supports static-sized tensors with a graph that has dynamic-sized tensors (tensor#-1 is a dynamic-sized tensor).
PS C:\Users\HP>
```

Selenium Test for Enrollment Form



```
try:
    # Open the enrollment form directly
    driver.get('https://nu.technuob.com/enroll/')

    # Wait for the form to load
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "enrollmentForm"))
    )

    # --- Personal Information Tab ---
    # Fill personal information with delays
    driver.find_element(By.ID, "firstName").send_keys("Lambodharan")
    time.sleep(1)
    driver.find_element(By.ID, "lastName").send_keys("Vinayagamoorthy")
    time.sleep(1)
    driver.find_element(By.ID, "nameWithInitials").send_keys("v.lambotharan")
    time.sleep(1)
    driver.find_element(By.ID, "nicPassport").send_keys("200084405583")
    time.sleep(1)
    dob_field = driver.find_element(By.ID, "dob")
    driver.execute_script("arguments[0].value = '2001-06-21';", dob_field)
    dob_field.click()  # Trigger any validation
    time.sleep(3)
    driver.find_element(By.XPATH, "//input[@name='gender'][@value='Male']").click()
    time.sleep(1)
    driver.find_element(By.ID, "permanentAddress").send_keys("44,Navalar-Road ,jaffna")
    time.sleep(1)

    # Select Province
    province = driver.find_element(By.ID, "province")
    province.send_keys("Northern")
    time.sleep(1)

    # Wait for District dropdown to populate
    WebDriverWait(driver, 5).until(
        EC.presence_of_element_located((By.XPATH, "//select[@id='district']/option[not(@disabled)]"))
    )
    driver.find_element(By.ID, "district").send_keys("Jaffna")
    time.sleep(1)

    driver.find_element(By.ID, "mobileNumber").send_keys("+94712345678")
    time.sleep(1)
    driver.find_element(By.ID, "secondaryMobile").send_keys("+94787654321")
```



Lambodharan

Vinayagamoorthy

Name with Initials *
v.lambotharan

NIC/Passport No *
200084405583

Date of Birth *
06/21/2001

Gender *
[✓ Male] Female

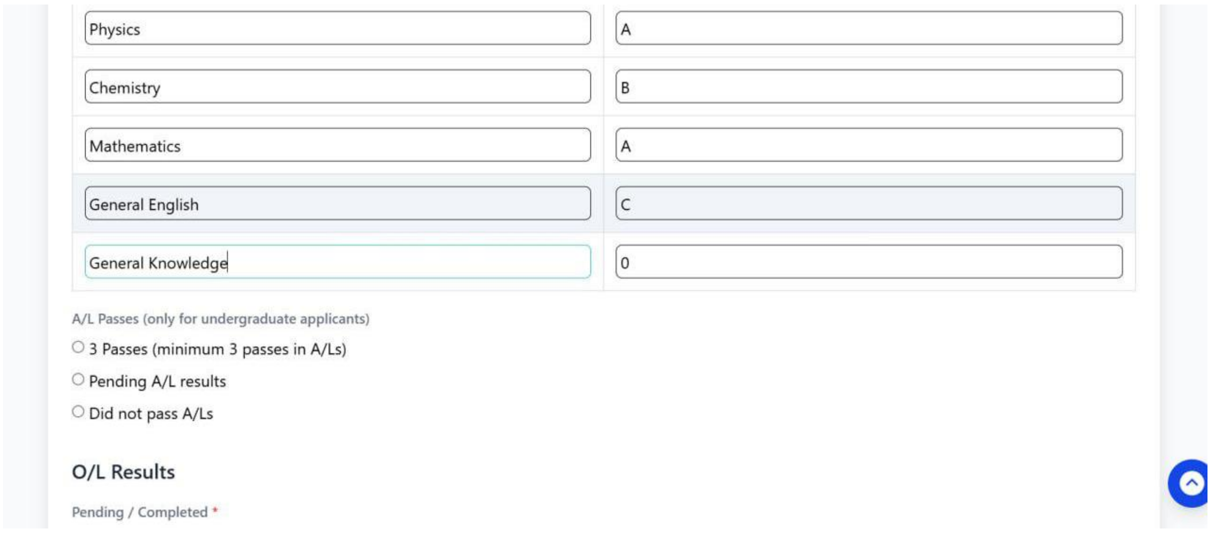
Permanent Address *
44,Navalar-Road ,jaffna

Province *
Northern

District *
Jaffna

Mobile Number *
+94712345678

Secondary Mobile Number



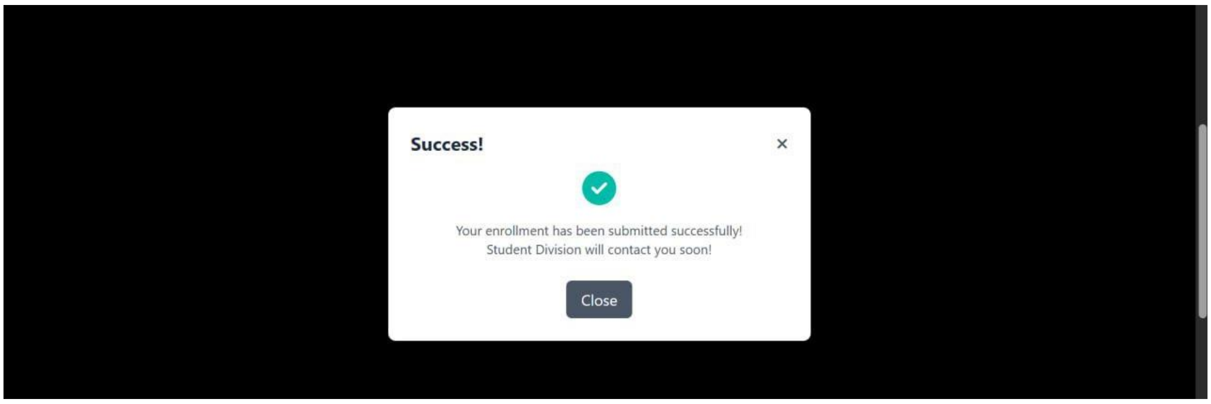
Physics	A
Chemistry	B
Mathematics	A
General English	C
General Knowledge	0

A/L Passes (only for undergraduate applicants)

○ 3 Passes (minimum 3 passes in A/Ls)
○ Pending A/L results
○ Did not pass A/Ls

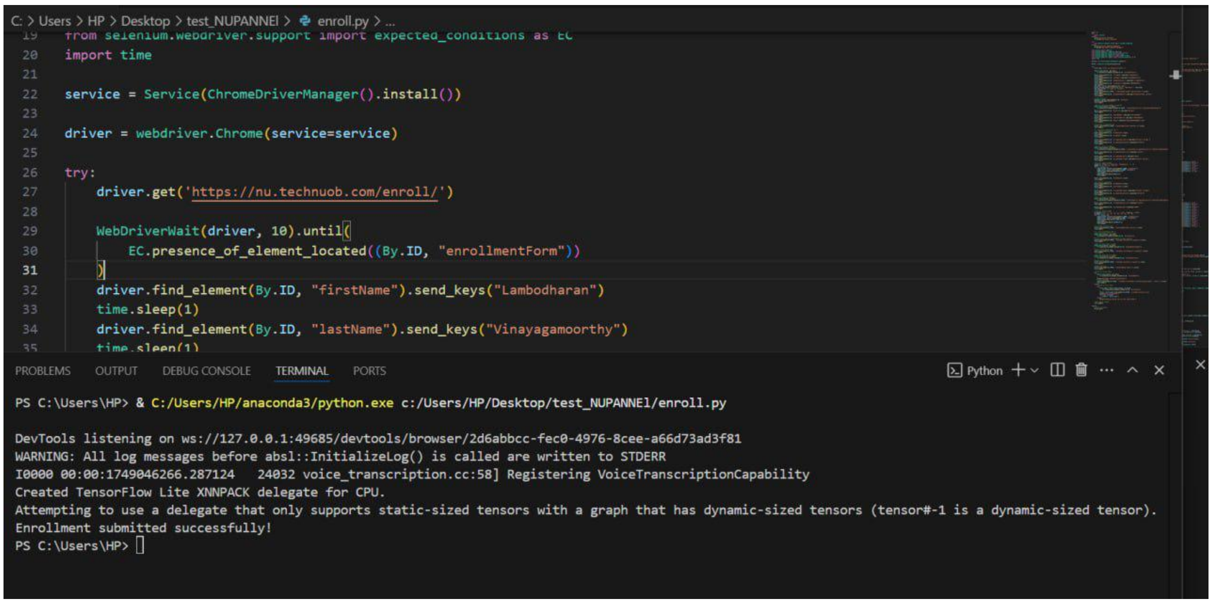
O/L Results

Pending / Completed *



Success!

Your enrollment has been submitted successfully!
Student Division will contact you soon!



```python
from selenium.webdriver.support import expected_conditions as EC
import time

service = Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=service)

try:
    driver.get('https://nu.technuob.com/enroll/')

    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "enrollmentForm"))
    )
    driver.find_element(By.ID, "firstName").send_keys("Lambodharan")
    time.sleep(1)
    driver.find_element(By.ID, "lastName").send_keys("Vinayagamoorthy")
    time.sleep(1)
```

```
PS C:\Users\HP> & C:/Users/HP/anaconda3/python.exe c:/Users/HP/Desktop/test_NUPANNEl/enroll.py

DevTools listening on ws://127.0.0.1:49685/devtools/browser/2d6abbcc-fec0-4976-8cee-a66d73ad3f81
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1749046266.287124   24032 voice_transcription.cc:58] Registering VoiceTranscriptionCapability
Created TensorFlow Lite XNNPACK delegate for CPU.
Attempting to use a delegate that only supports static-sized tensors with a graph that has dynamic-sized tensors (tensor#-1 is a dynamic-sized tensor).
Enrollment submitted successfully!
PS C:\Users\HP> 
```

6 Future Enhancement

In the future, we are planning to improve the SLIITNU Panel by adding more useful features. One of our main ideas is to include more graph-based visualizations in the admin dashboard. Right now, the admin can see some graphs showing student enrollment details. We plan to expand this by adding separate graphs for each feature, such as document uploads, payment status, and test registrations. These graphs will help the admin to clearly understand the data by week, month, or year.

We are also planning to train the chatbot with more information. Once the web application is fully hosted online, we will give the chatbot access to the real data and pages from this application. This will help the chatbot answer questions more accurately and guide students through the system more easily.

In addition, we will continue to provide updates and improvements to the system based on the feedback from the admin team. If they ask for new features or changes, we are ready to add them and keep the system updated. We also plan to upgrade the database in the future to a more powerful one like PostgreSQL for better performance when handling a large number of records.

7 Summary and Discussion

The SLIITNU Panel is a web-based system that we developed to help Northern Uni to manage student registration and other administrative tasks more easily. Before this system, the university used Excel sheets, notebooks, and printed documents. These manual methods were time-consuming and hard to manage**,** especially as the number of students increased.

With our new system, students can register online, upload documents, check their payment history, and read announcements. A chatbot is also available to answer basic questions. For the admin team, the system provides dashboards to manage student data, send messages via Telegram and email, and download data as Excel or PDF files. Additionally, the system offers many more features for both admin and student categories.

We used Django and Wagtail for the backend, and Tailwind CSS and HTML for the frontend. The system was tested on different devices, and we followed the Agile method, which helped us build the system step by step. Each feature was created based on feedback from the university staff, so it matches their real needs.

During this project, we learned how to work as a team, gather user requirements, and build a useful web application. The system helps reduce manual work, saves time, and improves overall organization. In the future, we plan to add more features like detailed graphs, a smarter chatbot, and regular updates based on admin requests. Overall, the SLIITNU Panel will support Northern Uni in doing their administrative work in a faster and more organized way.

References

[1] Schafer, M., "Django by Example: Build powerful and reliable Python web applications from scratch," 2nd ed., Packt Publishing, 2021. [Accessed 10 March 2025].

[2] Overland, B., "Learning Wagtail: Building modern content-managed websites with Wagtail CMS," Apress, 2018. [Accessed 12 March 2025].

[3] Silberschatz, A., Korth, H.F. and Sudarshan, S., "Database System Concepts," 7th ed., McGraw-Hill Education, 2019. [Accessed 14 March 2025].

[4] S. P. Adithela, M. Christie, S. Marru, and M. Pierce, "Django content management system evaluation and integration with apache airavata," in *Proc. Practice and Experience on Advanced Research Computing: Seamless Creativity*, 2018, pp. 1-4.

[5] S. Chen, S. Ahmmed, K. Lal, and C. Deming, "Django web development framework: Powering the modern web," *Am. J. Trade Policy,* vol. 7, no. 3, pp. 99–106, 2020.

[6] J. Forcier, P. Bissex, and W. J. Chun, *Python Web Development with Django*. Boston, MA: Addison-Wesley Professional, 2008.

[7] A. Chandiramani and P. Singh, "Management of Django web development in Python," *J. Manage. Serv. Sci. (JMSS),* vol. 1, no. 2, pp. 1–17, 2021.