28/12/2021

PROJECT REPORT

E- KYC

Project name: NSU E-KYC

Submitted by:

Md. Azmi Siddique 1812307042
Md. Imtiaz Haque 1811330042
Zarin Tasnim 1831423642
Md. Najmus Sakib 1813345642

North-South University, Dhaka, Bangladesh

Email: azmi.siddique@northsouth.edu, imtiaz.haque@northsouth.edu , zarin.tasnim21@northsouth.edu, najmus.sakib@northsouth.edu

Repository link : https://bitbucket.org/nabeel_mohammed/299.11.5/src/master/

Glossary of terms

Algorithm

An unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing, and automated reasoning tasks. They are ubiquitous in computing technologies.

Application programming interface (API)

A set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

Database

An organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex, they are often developed using formal design and modeling techniques.

Internet

The global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.

Library (computing)

A collection of non-volatile resources used by computer programs, often for software development. These may include configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values, or type specifications.

Programming language

A formal language, which comprises a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms.

Python

Is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Software testing

Is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Source code

In computing, source code is any collection of code, with or without comments, written using a human-readable programming language, usually as plain text. The source code of a program is specially designed to facilitate the work of computer programmers, who specify the actions to be performed by a computer mostly by writing source code. The source code is often transformed by an assembler or compiler into binary machine code that can be executed by the computer. The machine code might then be stored for execution at a later time. Alternatively, source code may be interpreted and thus immediately executed.

User

person who utilizes a computer or network service. Users of computer systems and software products generally lack the technical expertise required to fully understand how they work. Power users use advanced features of programs, though they are not necessarily capable of computer programming and system administration.

List Of figures 5

List Of tables 5

Executive Summary 6

Introduction 7

System Features 8

UML Use case diagram 8

Expanded Use Case 9

List and Description of API 10

System Design 11

Architectural Pattern 11

High Level Components and Interactions 12

Database Design 13

UML class diagram 14

Tools and External Libraries: 15

Testing 15

Installation Instruction 15

Conclusion: 21

Appendix 22

List of figures

1. UML Use Case Diagram……..8
2. Database Design…….13
3. UML class diagram……14

List of tables

1. List and Description of API……...10

2. Test Report…….12

Executive Summary

Through identity data acquisition, NSU E-KYC provides information about NSU students and operates as a connector between students and customers. The glossary terminology, table of contents, tables, and figures are all included in this project report. The features of NSU E-KYC are analyzed and discussed, as well as the design and implementation process of each element. Each section describes the strategy used to accomplish the project objectives. The project's required flow is specified in the report structure. The Unified Modeling Language Use Case Diagram and the Expanded Use Case, as well as a list and explanation of application programming interfaces, are included in System Features. The Architectural Pattern, as well as descriptions of the High Level Components and their Interactions, will be included in the System Design. Database Design includes the organized data according to the database model. It will have a UML class diagram in it. Python, SQLite3, Django, and other tools and external libraries were utilized in this project. We tested the website after finishing the design and development, and the test technique is supplied with the report for each element. Each step is accompanied by installation instructions. The project demonstration portion follows, followed by the conclusion. In NSU E-kyc, information is stored about students' grades for each semester, as well as course information and course specifics, and allows administrators to add, update, and delete information. The View option comprises each student's personal information, which is required for ekyc customers to verify the students of North- South university. It incorporates the API required to limit customer access to particular student information. Face recognition technology is being used in the project. The system's administrator will be able to log in and log out. In this project, the design structure, strategies used, tools, and application will be addressed in the first section, which will focus on the product specifics, work, and use. The design structure, strategies used, tools, and application will be discussed gradually.

Introduction

NSU-EKYC establishes a link between North-South University students and visitors to this website. Customers will have access to specific information about the students and will be able to verify the individuals using this feature. It contains the student's personal information as well as academic information. Face recognition technology will be able to discover and verify students, according to the needs of the customer. It has an API that allows the admin to limit the customer's access. For the design of the website we used the framework Django and used python as the programming language. We used sqlite3 as the database. We created an admin dashboard that includes student information, Information on the Course, Information about the customer, Page with details about the students, Page with details for customers, Page with details on the course. Each section has a curd view feature where the admin can edit, add, delete. We designed the student information in such a way that administrator can add grades and the system will compute them according to NSU grading regulations. The elements of the system will be discussed in the second chapter, as well as an UML use case diagram with extensive information on the project and extended use cases. Also, in the third chapter, we will go over the APIs that were used in this project, as well as the system design, which will be thoroughly detailed. Along with the database design, we'll look at the Architectural Pattern of the project, it’s High Level Components and their Interactions. The tools utilized in this project will be discussed in the fourth chapter. As an example, we used Django as a framework. It will also contain the information r of other tools and libraries used in this project. The testing section of the fifth chapter will include testing findings and strategies. The sixth chapter contains details on the project's whole installation process, while the seventh chapter contains a project demonstration portion. Finally, the project will be concluded in the eighth section.

System Features

a. UML Use case diagram

Fig: Use case UML Diagram

b. Expanded Use Case

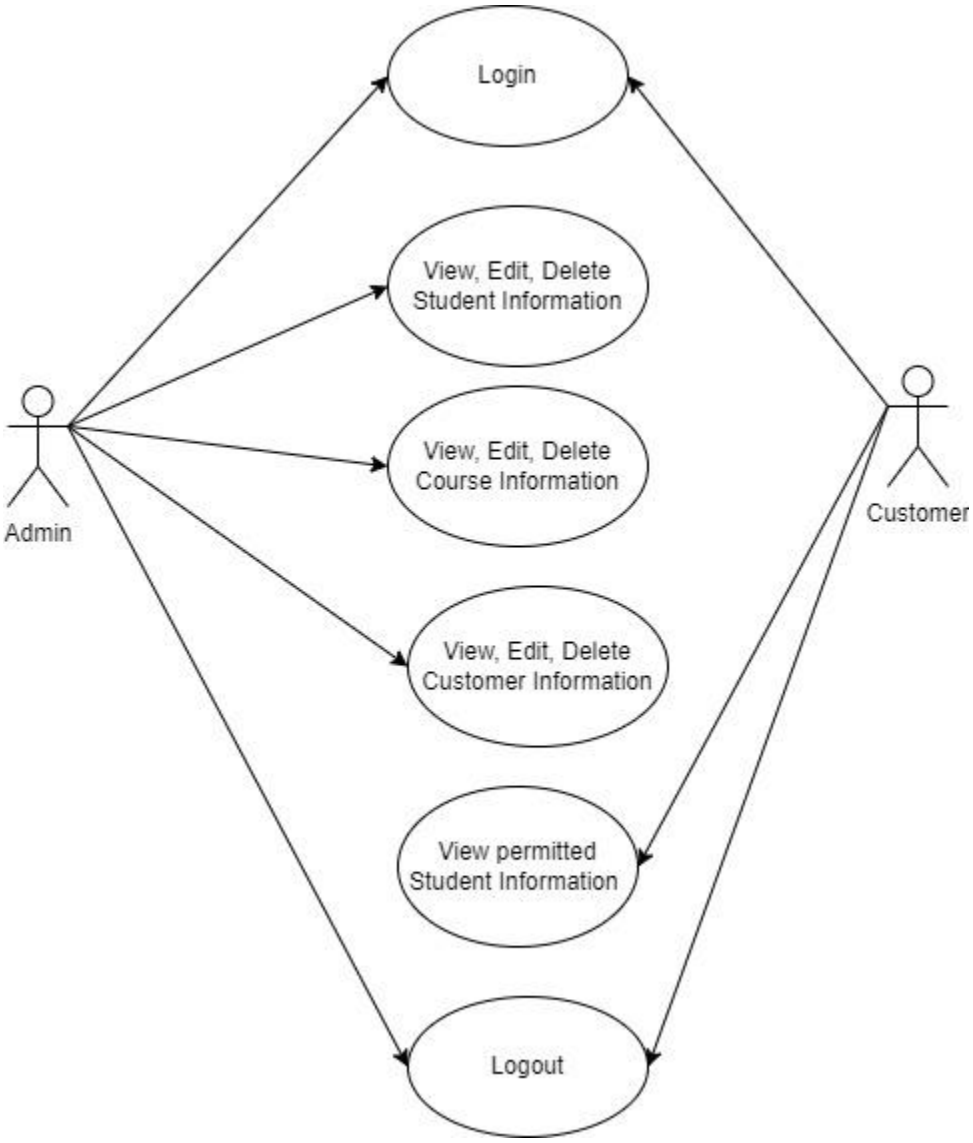

Fig: Expanded Use Case Diagram

a. List and Description of API

List	Description	Parameter	URL
viewstudent	In this API customer can view the student information according to the permission given by the admin. Such as an example a hospital wants to know about only the medical information and blood group so admin will only approve these permissions to view those details.	Only view the permitted details	http://127.0.0.1:8000/api/viewstudent/

System Design

a. Architectural Pattern

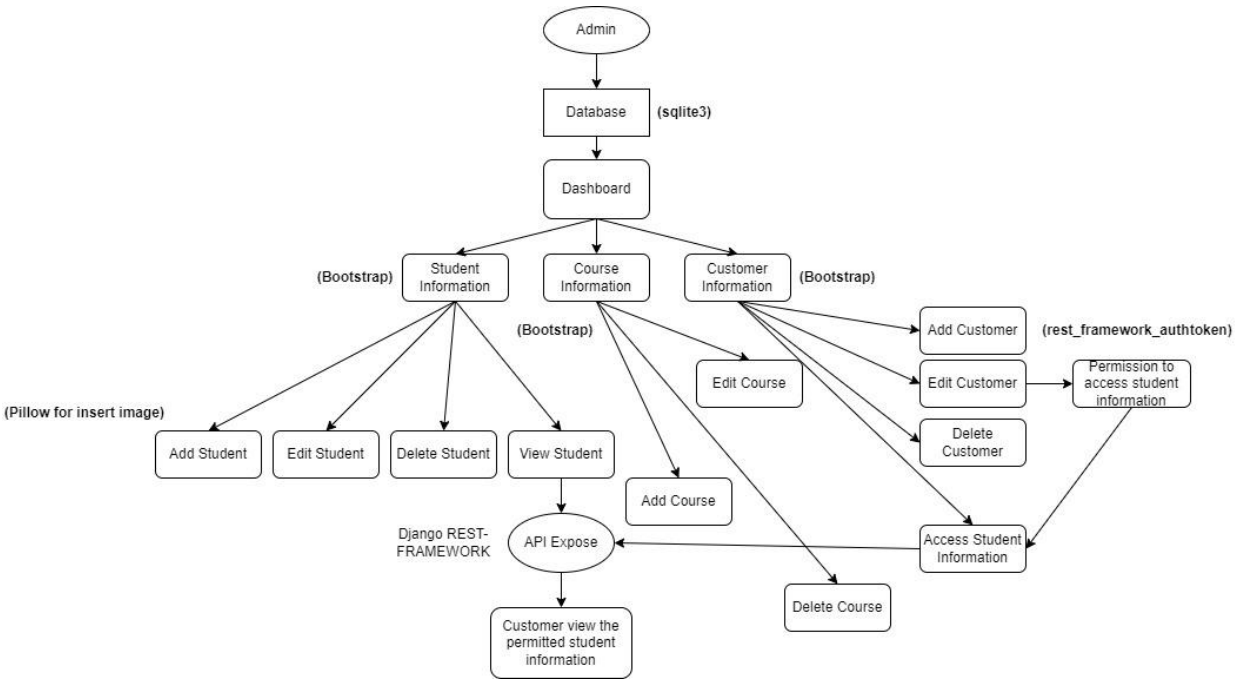

Fig: Architectural Pattern

We choose Django framework because it is an advanced python based framework and it has rapid development, fast processing, portable and it has clean design. The project contains an admin dashboard where we used sqlite3 as a database. The dashboard uses the bootstrap to display student information and we use the sqlite3 database as it comes with the default Django project, Easy to migrate.
We used the CRUD operation for managing (insert, delete etc.) the process easily. On adding the student information to the uploaded image we use python library pillow because it is effective and open source.
We have an API expose in the view area where we used the Django rest framework. API serves as a barrier to entry for customers who merely want to obtain student information. The customer has access to student data. We used token authentication because it gives customers a unique key while the admin added them.

b. High Level Components and Interactions

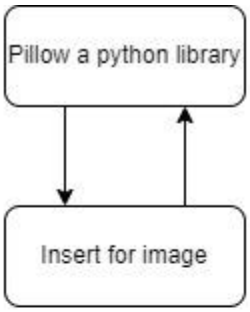

Fig: For inserting image

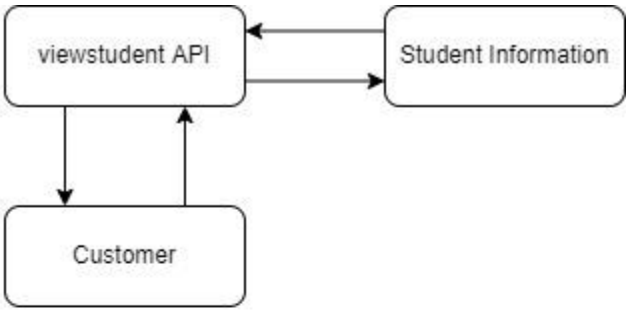

Fig: Access data with API

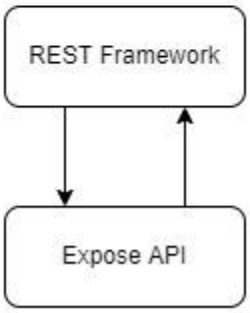

Fig: Expose API with REST Framework

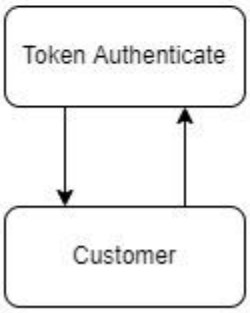

Fig: Generate token with adding a customer

c. Database Design

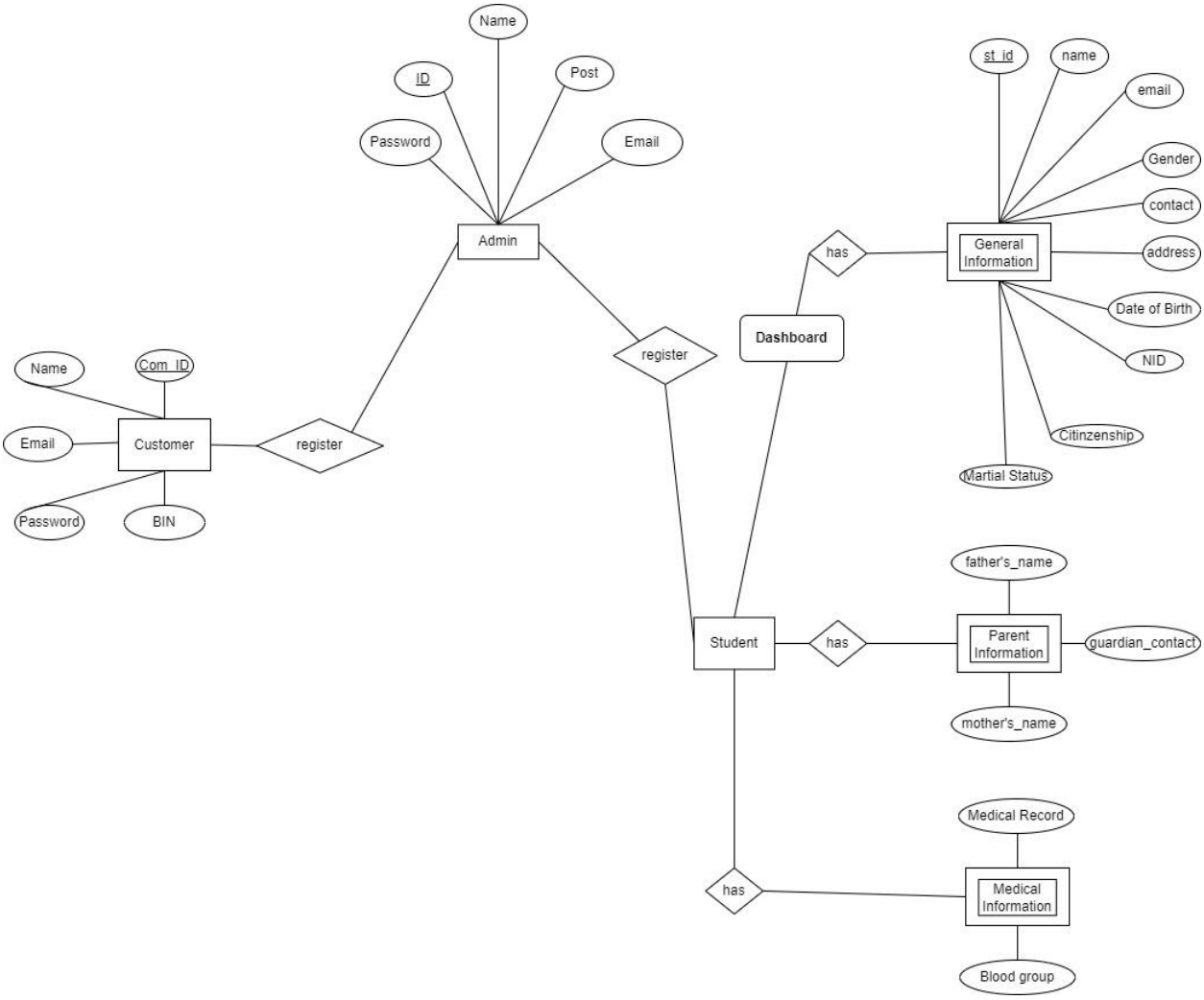

Fig: Database Design

d. UML class diagram

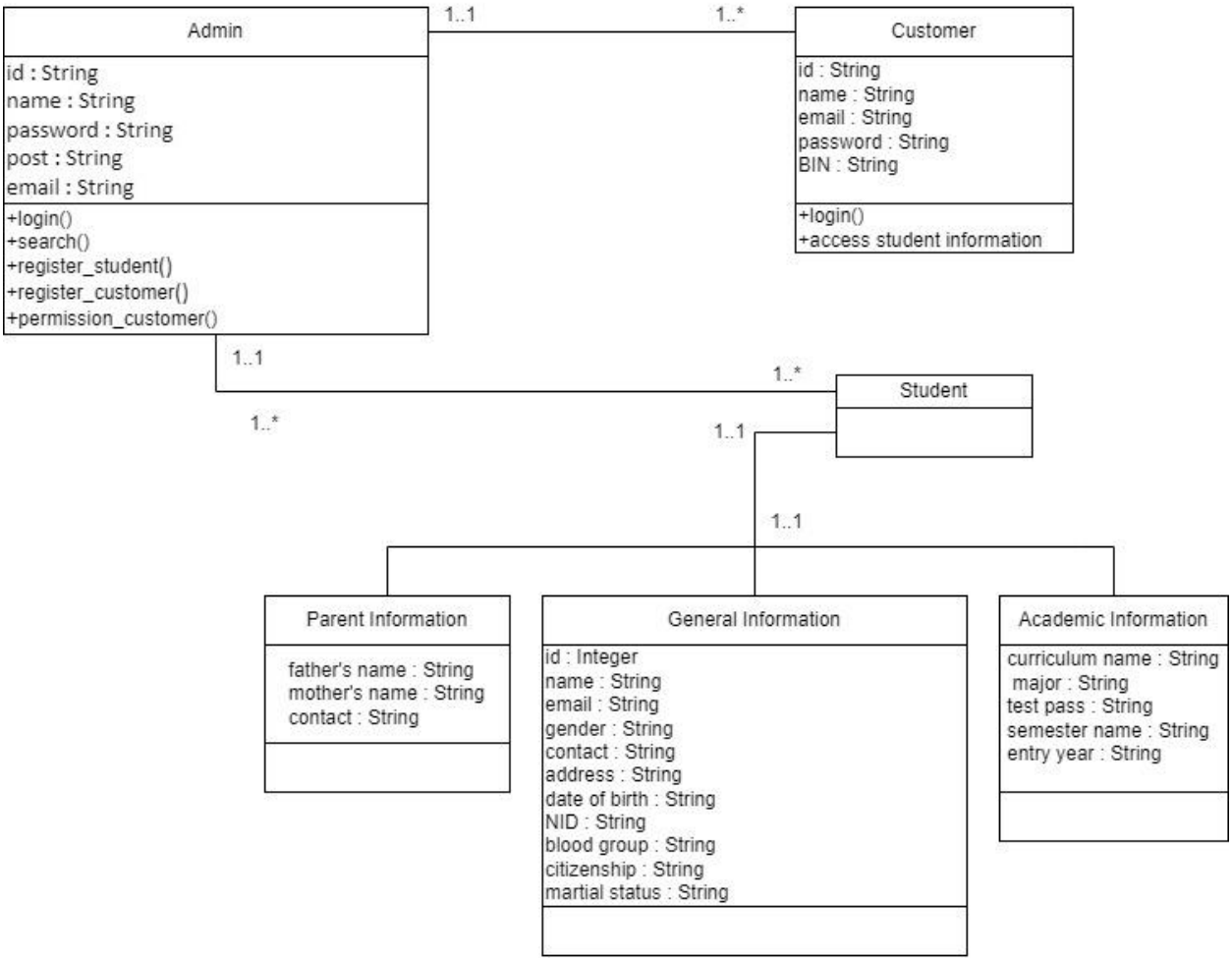

Fig: UML Diagram

Tools and External Libraries:

1. Database: Sqlite3
2. Python
3. API: Django rest framework API
4. Framework: Django
5. Bootstrap5

Testing

- Testing Strategy

  We manually input all the parameters and the result is shown below

- Test Report

Functionalities	Input	Test Results
Login	Email: admin@admin.com Password: admin	SUCCESSFUL
Add Operation	Student information, Customer information, Customer information	SUCCESSFUL
API	View student information	SUCCESSFUL
Log-out	Log out button	SUCCESSFUL
Grade	Academic information	UNSUCCESSFUL
Face resignation	image	UNSUCCESSFUL

Installation Instruction

Clone the repository
Go to admin
Write cmd
Install: pip install pillow
Type py manage.py runserver
Copy link: http://127.0.0.1:8000/
Run in browser

Admin log-in:

- Email: admin@admin.com
- Password: admin

Product Demonstration

- Log-in

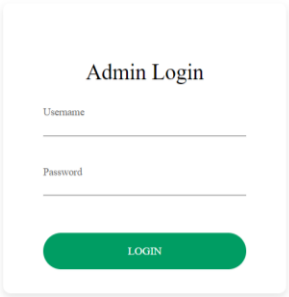

Fig: Admin login page

- **Dashboard**

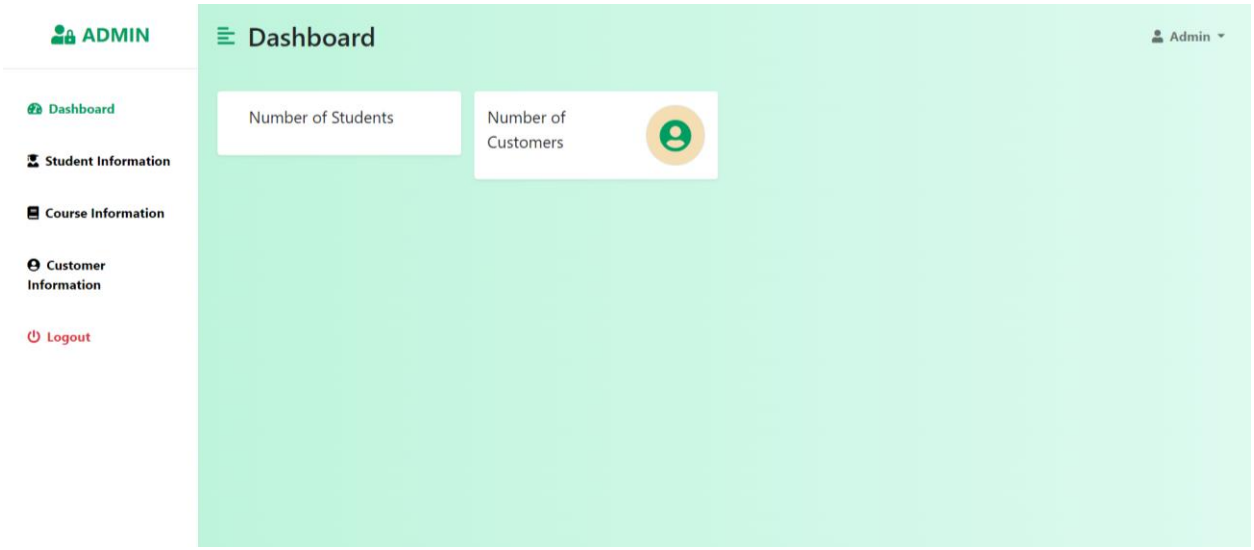

Fig: Dashboard

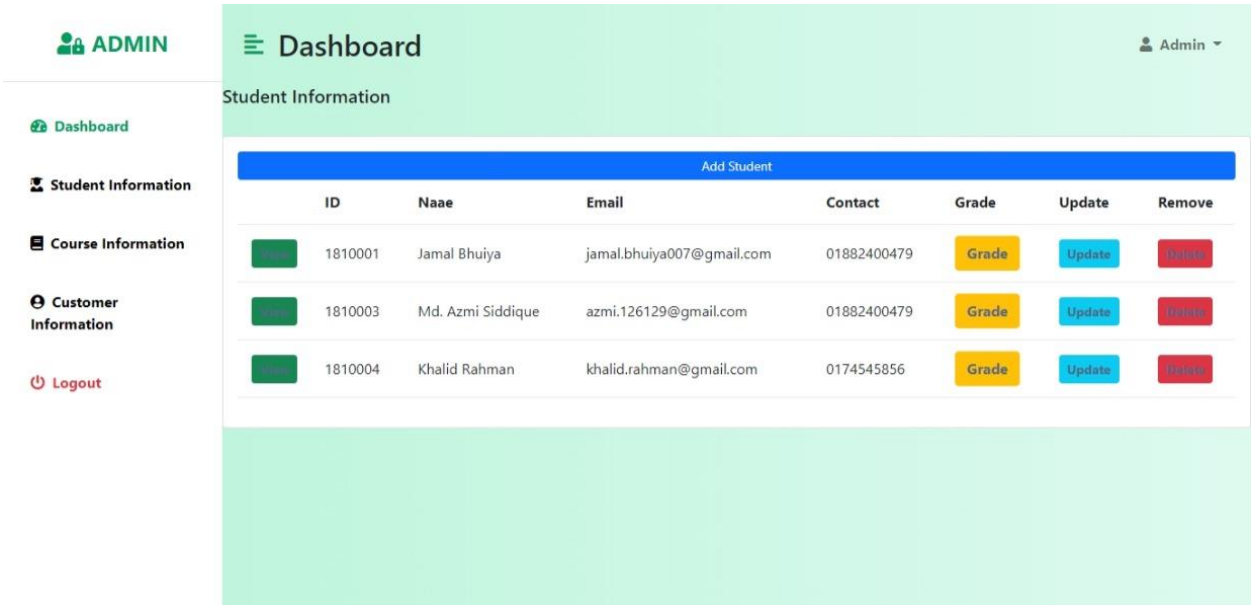

Fig: Student information dashboard

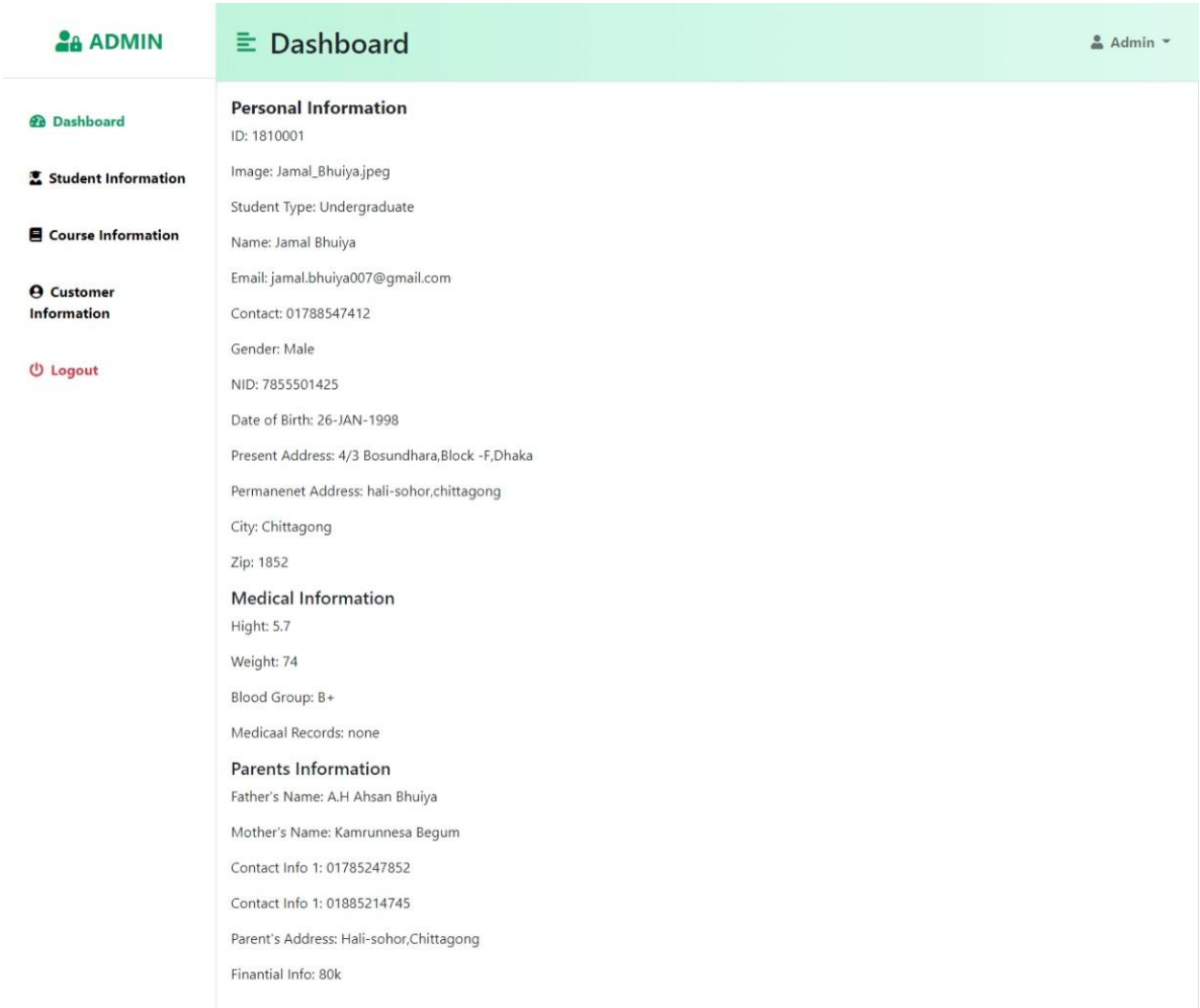

Fig: View Individual student information

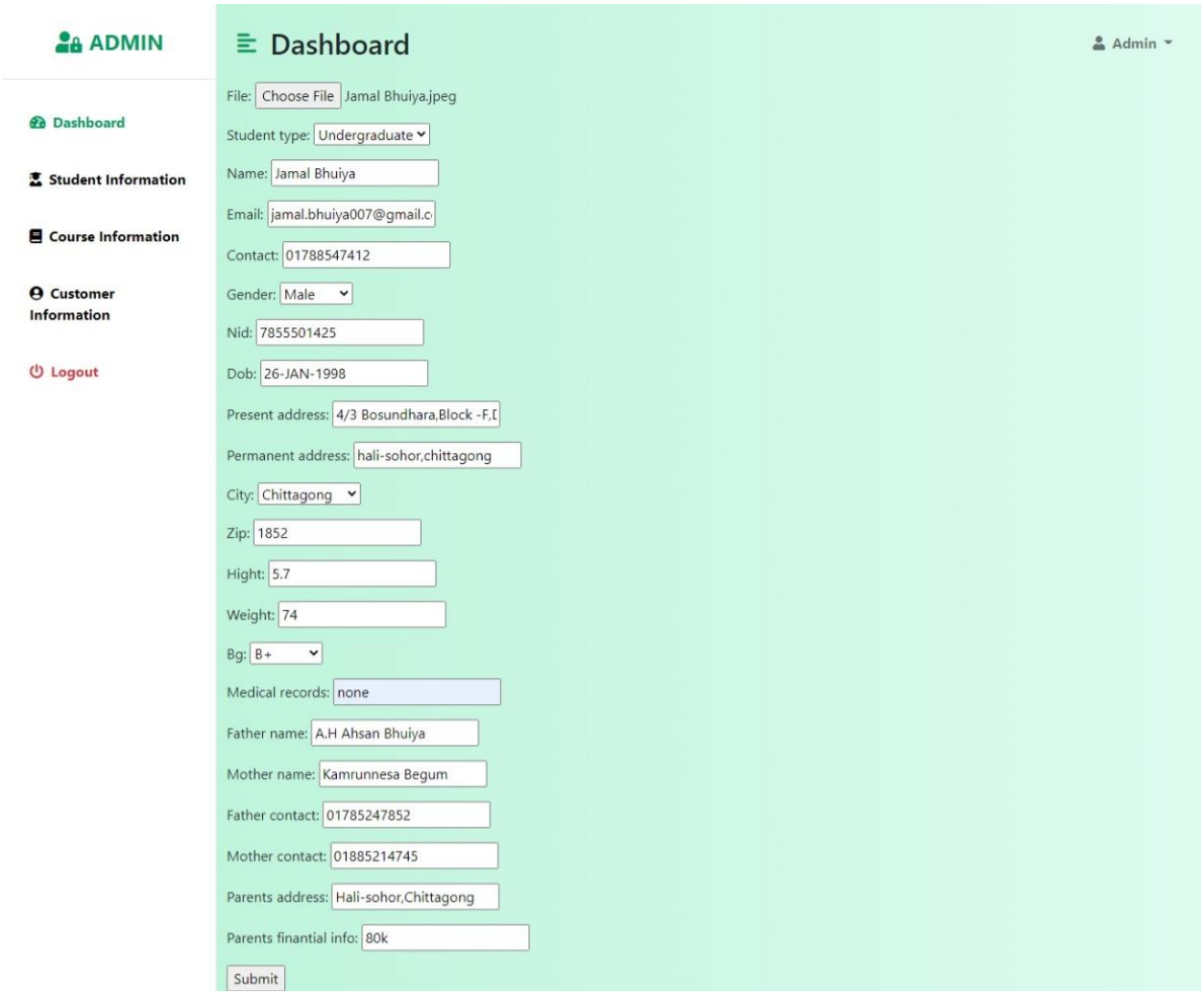

Fig: Register new student

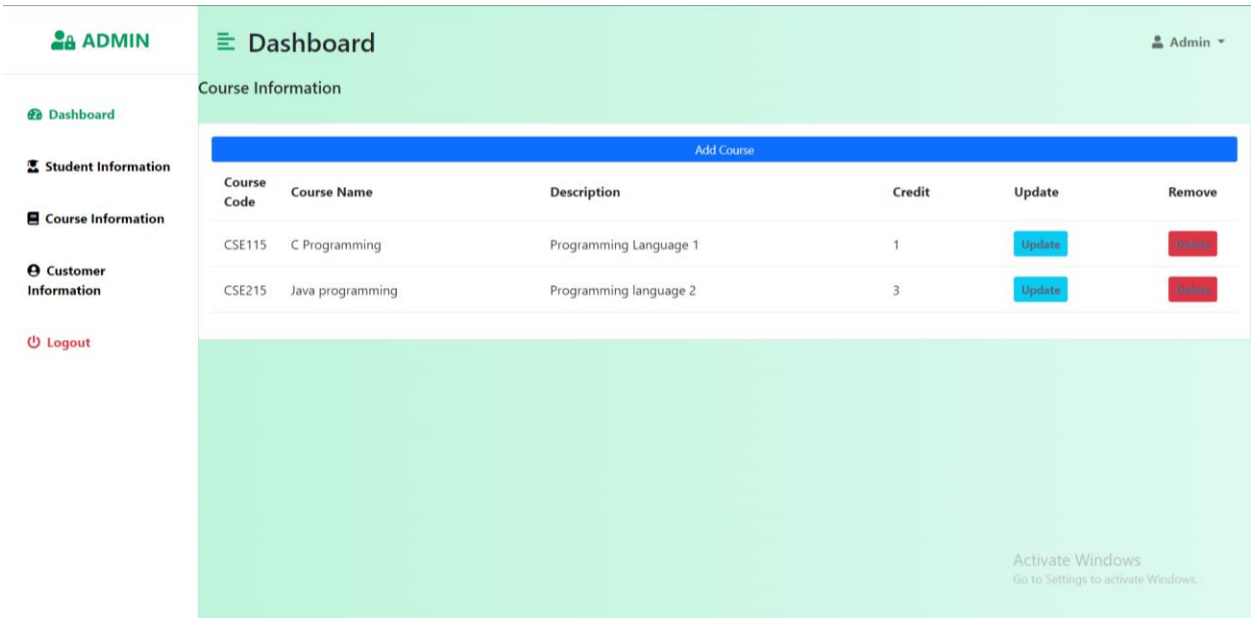

Fig: Course Information

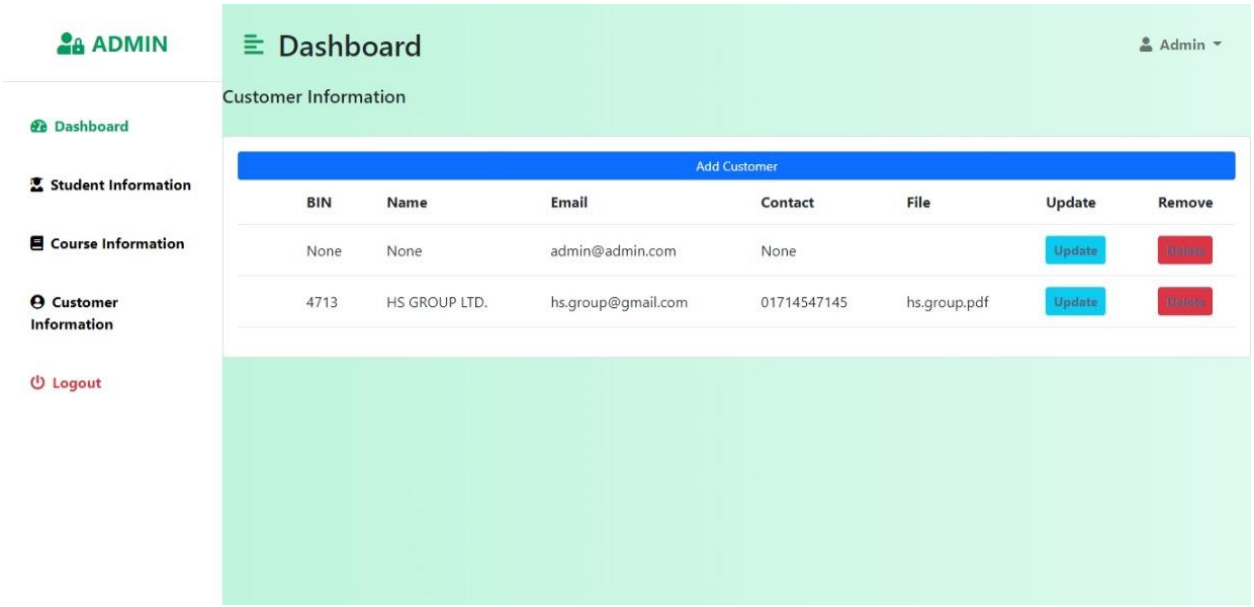

Fig: Customer information

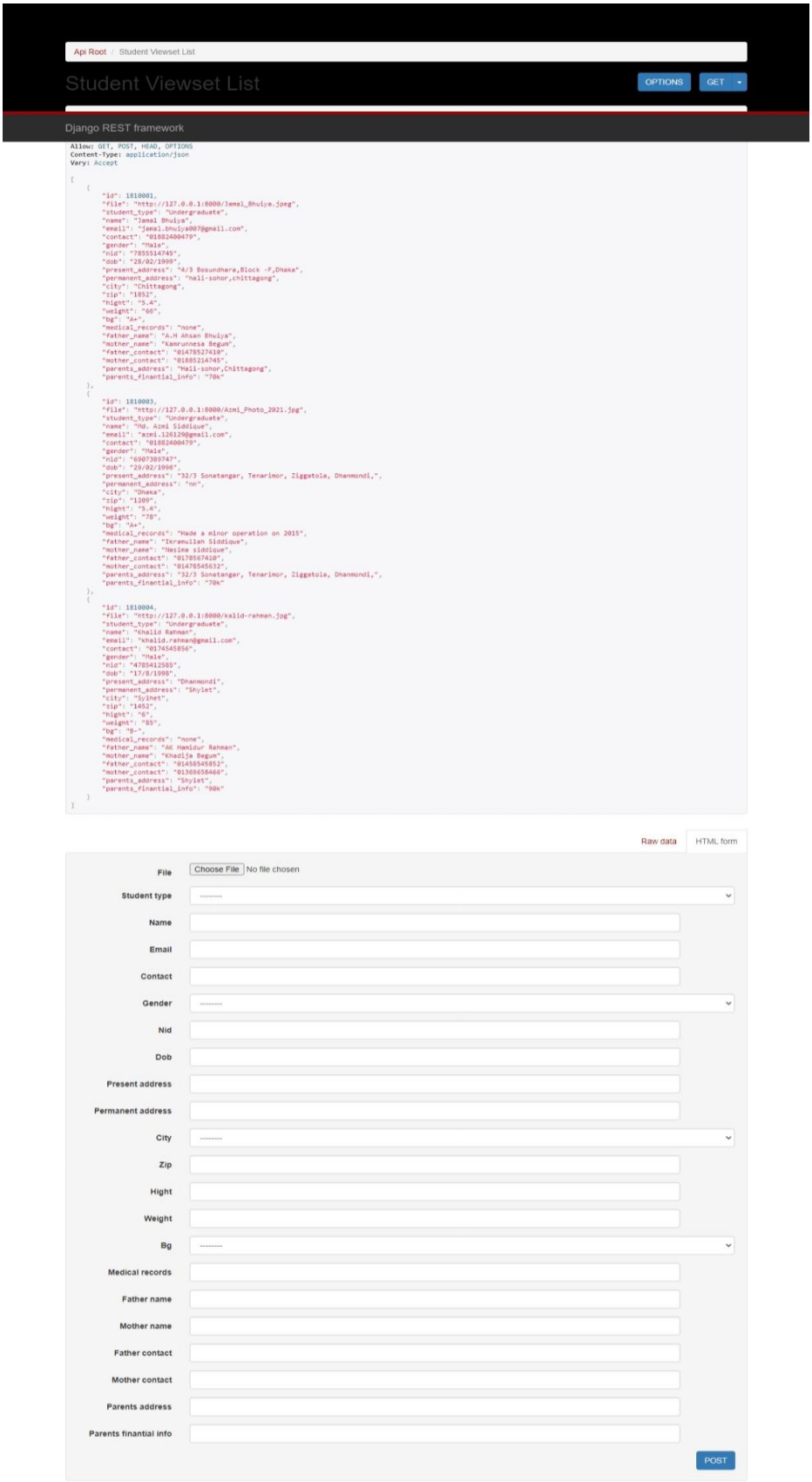

Fig: API to access student information

Conclusion:

NSU E-KYC is a significant tool for connecting and verifying individuals for a number of institutions. It establishes a connection and accomplishes a variety of functions. Customers will have access to detailed information on the students and will be able to use this tool to verify the individuals. It contains both personal and academic information about the student. Face recognition technology will be able to find and verify students based on the customer's requirements. It provides an API that allows the administrator to control the level of access given to customers. NSU E-KYC provides information on NSU students and acts as a link between students and customers by collecting identity data

Appendix

Contact information

Contact Person's Name: Md. Azmi Siddique 1812307642
Md. Imtiaz Haque 1811330642
Zarin Tasnim 1831423642
Md. Najmus Sakib 1813345642

Email: azmi.siddique@northsouth.edu, imtiaz.haque@northsouth.edu , zarin.tasnim21@northsouth.edu, najmus.sakib@northsouth.edu

North- South University, Dhaka, Bangladesh