# DEVELOPMENT OF

**NSTU WEBSITE WITH INTEGRATED CHATBOT**

## By

**Md Al Adnan**

Session: 2017-18 Roll: MUH1825008M

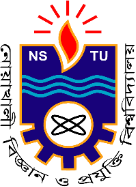
Software Engineering, IIT, NSTU

## Supervised By

**Tasniya Ahmed**

Assistant Professor Institute of Information Technology

Noakhali Science and Technology University

### Institute of Information Technology

**Noakhali Science and Technology University**

**Date of Submission 13 June 2023**

**DECLARATION**

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award or other institutions.

### Signature :

### Supervisor : Md Al Adnan

### ID : MUH1825008M

### Date:

### DEDICATION

I dedicate this complete project to the Institute of Information Technology (IIT) at Noakhali Science and Technology University (NSTU).

I would like to express my heartfelt gratitude to my project supervisor and the co-supervisor for their continuous guidance, valuable insights, and encouragement throughout the development process. Your expertise and mentorship have been invaluable in helping me navigate the challenges and complexities of this project.

Furthermore, I extend my thanks to my fellow classmates who have collaborated and contributed their efforts to this project. Your dedication, suggestion, and collective input have made this endeavor a success.

Last but not least, I express my deepest appreciation to my family and friends for their unwavering support, understanding, and encouragement during this journey. Your belief in my abilities has been a constant source of motivation.

This project is a culmination of the combined efforts, knowledge, and inspiration provided by all those mentioned above. Without their support, this endeavor would not have been possible. It is with great satisfaction and gratitude that I present this complete project document as a testament to our collective commitment and hard work.

Thank you all for being a part of this journey.

*Md Al Adnan*

### ACKNOWLEDGEMENT

Foremost, I reveal our heart-felt thanks and gratefulness to Almighty Allah for His divine benison that made me possible to complete the final year project successfully.

I am really grateful to my honorable project supervisor **Tasniya Ahmed**, Assistant Professor, Institute of Information Technology, Noakhali Science and Technology University, for his amazing idea of this project, his deep knowledge and keen interest as my supervisor helped to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Mohammad Salim Hossain**, Director, Software Engineering Program, IIT, as well as to the **Final Year Project Committee-2021** and also to other honorable faculty members of the institute for their kind help to finish the project.

I extend my thanks to my classmates who have collaborated closely with me on this project. Your suggestions and collective efforts have contributed immensely to the success of this endeavor.

Furthermore, I would like to acknowledge the support and understanding of my family and friends. Your unwavering belief in my abilities and constant encouragement have been a driving force behind my motivation and determination to complete this project.

Lastly, I would like to express my appreciation to all the individuals who have provided assistance, whether technical or moral, during the course of this project. Your contributions, big or small, have helped shape the outcome of this work.

To all those mentioned above and anyone else who has played a part in this project, I extend my heartfelt gratitude. Your support and involvement have been instrumental in the successful completion of this web application project and the preparation of this project document.

Thank you.

*Md Al Adnan*

### TABLE OF CONTENTS

CHAPTER 1 1

* 1. [Project Overview 2](#_TOC_250094)
  2. [Project Purpose 2](#_TOC_250093)
     1. [Background 3](#_TOC_250092)
     2. [Benefits & Beneficiaries 3](#_TOC_250091)
     3. [Goals 4](#_TOC_250090)
  3. [Stakeholders 4](#_TOC_250089)
  4. [Scope and Limitations 4](#_TOC_250088)

[CHAPTER 2 5](#_TOC_250086)

* 1. [Project Charter and Goals 5](#_TOC_250085)
  2. [Project Timeline and Milestones 6](#_TOC_250084)
     1. [Gantt Chart 6](#_TOC_250083)
     2. [Release Plan 7](#_TOC_250082)
  3. [Resource Allocation and Budgeting 8](#_TOC_250081)
  4. [Risk Assessment and Mitigation Strategies 9](#_TOC_250080)

[CHAPTER 3 10](#_TOC_250079)

* 1. [Functional Requirements 10](#_TOC_250078)
  2. [Data Requirements 16](#_TOC_250077)
  3. [Performance Requirements 16](#_TOC_250076)
     1. [Speed And Latency Requirements 16](#_TOC_250075)
     2. [Precision & Accuracy Requirements 17](#_TOC_250074)
     3. [Capacity Requirements 17](#_TOC_250073)

CHAPTER 4 18

* 1. [Use Case Diagram 18](#_TOC_250041)
  2. [Use cases descriptions 19](#_TOC_250040)

[CHAPTER 5 30](#_TOC_250038)

* 1. [System Architecture Overview 31](#_TOC_250037)
  2. [Database Schema Diagram 32](#_TOC_250034)
  3. [User Interface Design (Techniques & Strategies) 33](#_TOC_250033)

CHAPTER 6 34

* 1. [Development Tools & Technology 34](#_TOC_250032)
     1. [User Interface Technology 35](#_TOC_250031)
     2. [Implementation Tools & Platforms 36](#_TOC_250026)

[CHAPTER 7 38](#_TOC_250019)

* 1. [Testing Features 39](#_TOC_250018)
     1. [Features to be tested 39](#_TOC_250017)
     2. [Features not to be tested 39](#_TOC_250016)
  2. [Testing Strategies 40](#_TOC_250015)
  3. [Testing Environment (hardware/software requirements) 41](#_TOC_250009)

CHAPTER 8 47

* 1. [User Manual (Cyber Center) 48](#_TOC_250007)
  2. [User Manual (Teacher) 49](#_TOC_250006)
  3. [User Manual (Student) 50](#_TOC_250006)

CHAPTER 9 51

* 1. [Github Link 51](#_TOC_250005)
  2. [Critical Evolution 51](#_TOC_250004)
  3. [Limitations 51](#_TOC_250003)
  4. [Future Scope 51](#_TOC_250001)

[References 51](#_TOC_250000)

**LIST OF FIGURES**

Figure – 2.2.1: Gantt Chart 6

Figure – 4.1.1: Use Case Diagram 18

Figure – 5.2: Database Schema Diagram 24

### LIST OF TABLES

The chatbot should be able to reply to user inquiries effectively 10

The chatbot should be handle ambiguous queries 10

Users may text their inquiries 11

Users can ask query via text 11

Users can ask query via voice 11

Users must receive text messages as response 11

Users must be receiving answer via voice 12

Users get login to the system 12

Users can register to the system 12

Cyber center must be editing feedback information 12

Cyber center must be storing feedback data 13

Cyber center can control about section 13

Cyber center manage administration section 13

Cyber center can update faculty information 13

Cyber center can manage academic information 14

Teachers can update their own profile 14

Cyber center updated the job circulars 14

Cyber center can manage notice 14

Cyber center can announce news and events 15

Any Research update will be announced by Cyber center 15

Cyber center can handle miscellaneous section 15

**LIST OF ABBREVIATION**

IIT - Institute of Information Technology

NSTU - Noakhali Science and Technology University

SRS - Software Requirements Specification CSS - Cascading Style Sheets

HTTP - Hypertext Transfer Protocol HTML - Hypertext Markup Language

API – Application Programming Interface UI – User Interface

SDLC – Software Development Life Cycle SQL – Structured Query Language

# CHAPTER 1 INTRODUCTION

The first chapter gives an overview of the online application "NSTU Dynamic Website with Integrated Chatbot" created a modern online platform that embodies the vibrant spirit and endless possibilities offered by our renown university. It provides us many sections from where users can know about any small details about NSTU, like about Cyber centeristration, about academic information, admission related information, notice board from where users get update frequently, besides users can get updated news and events. From integrated chatbot we can get answers by asking which is reply to us in static way. It uses natural language processing (NLP) techniques and artificial intelligence (AI) algorithms to understand user queries and provide relevant responses or perform actions. In this Website has Cyber center panel were Cyber center plays Cyber center role. Rather Cyber center can add teachers as a moderator to edit their personal profile. This chatbot receives questions from users, tries to understand the question, and provides appropriate answers. Our dynamic website serves as a reflection of our commitment to providing a cutting-edge educational environment where students, faculty, staff, and visitors can seamlessly connect, engage, and explore. It does this by converting Bangla and English sentence into a machines-friendly query, then going through relevant data to find the necessary information and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer. For example, it will receive the question “IIT means?”, it will give a response “IIT Stands for Institute of Information Technology”. The goal is to provide chatbot for users a quick and easy way to have their questions answered.

NSTU Dynamic Website with Integrated Chatbot will help to know any information related with NSTU. It can provide quick and accessible support to students, rather than chatbot give the answers to their frequently asked questions. This website gives every update about Cyber centeristration, academic, admission, notice, news and events and also research. If users need any answers by asking questions so they can ask questions in chatbot, it will provide proper answer. Here Cyber center panel can edit and update everything they want, Cyber center also give a email and password for teachers log in such that they can join as a moderator. Edit and updated section will handle by the Cyber center panel. The website and chatbot are available 24/7, you just need to visit the site and access the site. This section tries to familiarize readers with the project's overview, purpose and scope.

#### Project Overview

The " NSTU Dynamic Website with Integrated Chatbot " web application is made to make it easier for students and non-students. This website helps students by giving results of their searching queries about NSTU related. Besides chatbot can give answers for every individual question, as a result user can save his valuable time. Websites will work dynamically and chatbot will understand human readable questions and give the best possible answers.

#### Project Purpose

The "NSTU Dynamic Website with Integrated Chatbot" project's goal is to build a reliable and approachable platform for a university or other educational institution. With this project, the institution will be able to manage notices, academic menus, student profiles, and faculty profiles in an effective and engaging manner. The following are the project's main goals:

* Dynamic Notices: The website will give Cyber centeristrators real-time access to notice creation, editing, and publication. This feature makes it possible for teachers and students to quickly and conveniently share critical information with one another, such as announcements, deadlines, and forthcoming activities.
* Academic Menu: The website will have a part devoted to the management of subject syllabi, course offerings, and other academic-related data. Students will have access to the menu to browse the courses that are offered, their descriptions, and any connected materials.
* Student Profiles: A separate area of the website will be dedicated to student profiles. Each student will have a profile page where they may access and edit their personal data, academic history, class schedules, and other pertinent information. By allowing them access to their academic records and encouraging connection with staff members, this feature attempts to empower students.
* Instructor Profiles: The website will contain a part for instructor profiles, similar to the area for student profiles. Each faculty member will have a personal profile page where they can manage their contact information, educational background, professional experience, and other information. This tool encourages openness and efficient communication within the organization by allowing students and Cyber centeristrators to browse teacher profiles.
* Integrated Chatbot: The website will have a built-in chatbot that will give employees and students an engaging and user-friendly interface. The chatbot will help visitors use the website by guiding them, responding to their commonly asked queries, and delivering pertinent information whenever they need it.

##### Background

As a fresher, a student or his guardian wants to know about the new varsity. Sometimes they go to the registered office for information or sometimes they call the helpline number, but it costs huge time consuming also does not find lot of information. Using this website and chatbot response according to student's need, he/she can frequently search anything about NSTU from this website and ask NSTU related question and information within a few seconds from chatbot. This website helps the users to find any information and asks questions the chatbot to fix his problem in proper way. Any tiny information can know from this website. Therefore, this project is focusing on providing a solution to this problem by giving the proper and accurate information about the whole campus to the users. This project aims to build a dynamic website and an integrated chatbot for NSTU to answer every person who asks about the university, facilities, admission information and policy.

##### Benefits & Beneficiaries

The "NSTU Dynamic Website with Integrated Chatbot" web application project offers several benefits and serves as a valuable tool for various beneficiaries.

##### Benefits:

* + - * Save effort and time: This website saves a lot of time for both university Cyber center and users. Also users just ask the question using natural languages chat bot reply to the most appropriate answer.
      * Provide detailed information about NSTU and Admission queries.
      * Easy Access to information.
      * Website and chatbot is available 24/7

##### Beneficiaries:

* + - * Students: Students who wish to enroll at Noakhali science and technology university and want to know about NSTU facilities, also existing students they can manage their own profile
      * Outside users: guardians or any persons who want to know about NSTU Facilities.
      * Teachers: Teacher can updates their profile information.

##### Goals

The "NSTU Dynamic Website with Integrated Chatbot" web application project aims to achieve the following:

* + - * Save time and effort for users and Cyber centeristrations.
      * Share NSTU related information.

#### Stakeholders

The stakeholders involved in the "NSTU Dynamic Website with Integrated Chatbot" web application project include:

* Students: They play a key role in “NSTU Dynamic Website with Integrated Chatbot” software projects. They benefit from websites and chatbot like they can know all unknown questions like NSTU Lab Facilities, Hall office, Central library, or admissions query.
* Teachers: They play another key role in “NSTU Dynamic Website with Integrated Chatbot” software projects. They benefit from websites and chatbot like they can know all unknown questions like NSTU Lab Facilities, also update their information as well.
* Outside user: non-Students means they can be anyone, like they can be guardians on any persons who want to know about NSTU University.
* Cyber centeristrators: Cyber centeristrators can review the feedback of users and store the feedback for future training, also update full management. Cyber center play Cyber centeristrators rule here.

#### Scope and Limitations

The scope and limitations of the " NSTU Dynamic Website with Integrated Chatbot " web application project are as follows:

##### Scope

* Save time and effort for users.
* Provide quick and accessible support to student by answering their frequently asked questions.
* Enhances the overall student experience and helps students make the most of campus resources.

##### Limitations

* Lack of huge data.
* Cannot manage staff information.
* Cannot manage office information.
* Cannot manage photo gallery dynamically

# CHAPTER 2

**PROJECT PLANNING AND MANAGEMENT**

#### Project Charter and Goals

Project Charter

Project Name: NSTU website integrated with chatbot Project Sponsor: N/A

Project Manager: Md Al Adnan

Project Start Date: 23 January 2023

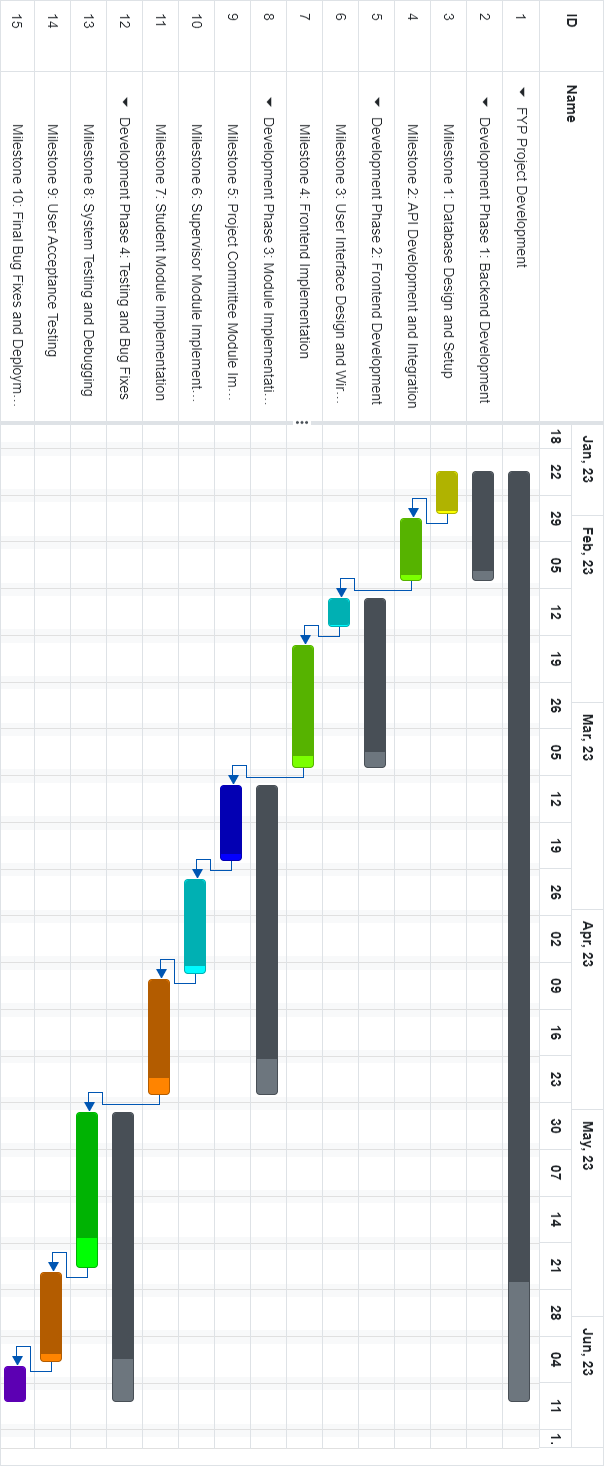
Project End Date: 13 June 2023 Project Goals:

* + - Develop a web application named " NSTU website integrated with chatbot " for the Noakhali Science and Technology University (NSTU).
    - Provide the best platform for students, teacher, outside users and cyber center.
    - Deliver a user-friendly interface that is intuitive and easy to navigate for all users.
    - Chatbot can assist with inquiries related to admissions, registration, course enrollment, campus facilities, financial aid, academic resources.
    - Website provides some important menu’s like about NSTU, academics, admission, news and events, publications.
    - Chatbot provides quick and accessible support to students by answering their frequently asked question.
    - Teacher and students can manage their own profile dynamically.

*Figure 2.2.1: Gantt Chart*

#### Project Timeline and Milestones

##### Gantt Chart



##### Release Plan

Version: 1.0

Release Date: 14 June 2023 Features Included:

##### Student Module

* + - * + Students can login to the system.
        + Students can ask queries.
        + Students can give feedback.
        + Students can update their own information.
        + Students can view all menus.

##### Cyber center Module:

* + - * + Cyber center can see the user’s feedback.
        + Cyber center can login to the system.
        + Cyber center can manage teachers information.
        + Cyber center can manage students information.
        + Cyber center can manage notice, events, news.
        + Cyber center can manage About nstu, Academic and Admission
        + Cyber center can store the feedback data.

##### Teacher Module:

* + - * + Teachers can login to the system.
        + Teachers can update their own information.
        + Teachers can view all menus.
        + Teachers can login to the system.
        + Teachers can ask queries.
        + Teachers can give feedback.

##### Release Goals:

* Provide a functional web application for users.
* Enable effective monitoring and maintenance of software projects.
* Ensure a user-friendly interface and smooth user experience.

##### Release Deliverables:

* Deployed web application accessible to users
* User documentation and guides for each module.
* Bug fixes and improvements based on user feedback.

#### Resource Allocation and Budgeting

##### Human Resources:

* + - * Development Team: 4 members (1 project manager, 2 developers, 1 designer)
      * Project Manager: 15 hours per week
      * Designers: 20 hours per week
      * Developers: 40 hours per week
      * Quality Assurance: 10 hours per week
      * Technical Support: As needed.

##### Infrastructure:

* + - * Hardware: High-performance workstation for each team member
      * Software:
        + IDEs: PyCharm’s
        + Version Control: Git
        + Project Management: JIRA
        + Communication: Slack, Zoom
        + Hosting: AWS or Digital Ocean
      * Development Environment: Python (NLP, FLASK)

##### Budgeting:

* + - * Development Costs:
        + Salaries: $X (estimated monthly salaries for team members)
        + Licenses: $Y (cost of any required software licenses)
      * Infrastructure Costs:
        + Hardware: $Z (estimated cost of high-performance workstations)
        + Software: $W (estimated cost of required software tools and subscriptions)
        + Hosting: $V (estimated monthly hosting costs on AWS or Digital Ocean)
      * Training Costs: $U (if any training or workshops are required)
      * Contingency Budget: 10% of the total budget
      * Total Project Budget: $X + $Y + $Z + $W + $V + $U + contingency budget

#### Risk Assessment and Mitigation Strategies

* + - Risk: Technical Challenges and Compatibility Issues

Mitigation: Conduct thorough feasibility studies and technical analysis before development. Perform comprehensive testing and debugging at each stage to identify and address compatibility issues.

* + - Risk: Insufficient User Acceptance

Mitigation: Involve users and stakeholders throughout the development process. Gather regular feedback, conduct usability testing, and incorporate suggestions to ensure user satisfaction.

* + - Risk: Security Vulnerabilities and Data Breaches

Mitigation: Implement robust security measures such as encryption, secure authentication, and regular security audits. Follow best practices for secure coding and conduct penetration testing.

* + - Risk: Scalability and Performance Issues

Mitigation: Perform load testing and optimization to ensure the application can handle increasing user loads. Implement caching mechanisms, database indexing, and other performance optimization techniques.

* + - Risk: Inadequate Documentation and Knowledge Transfer

Mitigation: Maintain comprehensive documentation throughout the development process. Conduct knowledge transfer sessions to ensure clear understanding among the development team and stakeholders.

* + - Risk: Budget Overruns

Mitigation: Regularly track and monitor project expenses. Conduct cost-benefit analyses and prioritize features to ensure effective budget management.

# CHAPTER 3

**SOFTWARE REQUIREMENTS SPECIFICATION**

#### Functional Requirements

A functional requirement is a specific capability or behavior that a system, software, or product must possess to fulfill its intended purpose and meet the needs of its users. These requirements describe what the system should do and how it should behave in terms of its functionality. Functional requirements focus on the tasks, operations, and features that the system should be able to perform. Here are functional requirements for “**NSTU website with integrated chatbot**.” Here table 1 shows the functional requirement that is “the chatbot should be able to reply to user inquiries effectively”, table 2 shows the functional requirement that is “the chatbot should handle ambiguous queries.”, table 3 shows the functional requirement that is “user may text their inquiries.”, table 4 shows the functional requirement that is “users can ask queries via voice.”, table 5 shows the functional requirement that is “users must receive text messages as responses.”, table 6 shows the functional requirement that is “users must be receiving answer via voice.”, table 7 shows the functional requirement that is “users can login to the system”, table 8 shows the functional requirement that is “Cyber center must be editing feedback information.”, table 9 shows the functional requirement that is “Cyber center must be editing feedback information.”, and table 10 shows the functional requirement that is “Cyber center must be storing feedback data.” table 11 shows the functional requirement that is “Cyber center can control about section” ,table 12 shows the functional requirement that is “Cyber center manage administration section” , table 13 shows the functional requirement that is “Cyber center can manage academic information”, table 14 shows the functional requirement that is “Teachers can update their own profile” , table 15 shows the functional requirement that is “Cyber center can manage notice”, table 16 shows the functional requirement that is “Cyber center can announce news and events” ,table 17 shows the functional requirement that is “ Any Research update will be announced by Cyber ”table 18 shows the functional requirement that is “ center Cyber center can handle miscellaneous section”

**Table 1: The chatbot should be able to reply to user inquiries effectively.**

|  |  |
| --- | --- |
| **FR-1** | The chatbot should be able to reply to user inquiries effectively |
| **Description** | When a user writes a message needing any queries, the chatbot understands the message and gives the user feedback. |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 2: The Chatbot should handle ambiguous queries.**

|  |  |
| --- | --- |
| **FR-2** | The Chatbot should handle ambiguous queries |
| **Description** | When a user enters an unclear message or lacks context, the system will replay an audio message instructing them to enter the correct information. |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 3: Users may text their inquiries.**

|  |  |
| --- | --- |
| **FR-3** | Users may text their inquiries. |
| **Description** | After login, the user can write a message for any queries in this message prompt. |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 4: Users can ask queries via voice.**

|  |  |
| --- | --- |
| **FR-4** | Users can ask queries via voice. |
| **Description** | After login, the user can deliver a voice message for any queries in this message prompt. |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 5: Users must receive text messages as responses.**

|  |  |
| --- | --- |
| **FR-5** | Users must receive text messages as responses. |
| **Description** | The user writes queries and sends them via message prompt, and after the system gets the message, the system replies answer via text |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 6: Users must be receiving answer via voice**

|  |  |
| --- | --- |
| **FR-6** | Users must be receiving answer via voice |
| **Description** | The user delivers a voice message, and after the system gets the message, the system replies answer via text |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 7: Users can log in to the system.**

|  |  |
| --- | --- |
| **FR-7** | Users can login to the system |
| **Description** | To login to the system, a user must write their email and password |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 8: Users can register to the system.**

|  |  |
| --- | --- |
| **FR-8** | Users can register to the system |
| **Description** | A user can register to the system using their email and password |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 9: Cyber center must be editing feedback information.**

|  |  |
| --- | --- |
| **FR-9** | Cyber center must be editing feedback information. |
| **Description** | When the system replies to a user’s message, the Cyber center can check and edit the feedback message |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 10: Cyber center must be storing feedback data.**

|  |  |
| --- | --- |
| **FR-10** | Cyber center must be storing feedback data |
| **Description** | Any responses provided by the user are recorded by the Cyber center for use in further training the model. |
| **Stakeholders** | Student, Non-student |
| **Priority** | High |

**Table 11: Cyber center can control about section.**

|  |  |
| --- | --- |
| **FR-11** | Cyber center can control about section |
| **Description** | Only Cyber center can update information about nstu, mission and vision of this varsity, rather all information related with university will handle by Cyber center. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 12: Cyber center can manage administration section.**

|  |  |
| --- | --- |
| **FR-12** | Cyber center manage administration section |
| **Description** | Cyber center can update the treasurer section and register section daily according to management system. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 13: Cyber center can update faculty information.**

|  |  |
| --- | --- |
| **FR-13** | Cyber center can update faculty information |
| **Description** | If any update or change is needed, so Cyber center can handle it. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 14: Cyber center can manage academic information.**

|  |  |
| --- | --- |
| **FR-14** | Cyber center can manage academic information |
| **Description** | If any update or change is needed, so Cyber center can handle it. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 15: Teachers can update their own profile.**

|  |  |
| --- | --- |
| **FR-15** | Teachers can update their own profile |
| **Description** | Cyber center give an email and a password for every teacher and they login to update their profile. |
| **Stakeholders** | Teachers, Cyber center |
| **Priority** | High |

**Table 16: Cyber center updated the job circulars.**

|  |  |
| --- | --- |
| **FR-16** | Cyber center updated the job circulars |
| **Description** | If any job post is needed to be announced, then Cyber center can update it. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 17: Cyber center can manage notice.**

|  |  |
| --- | --- |
| **FR-17** | Cyber center can manage notice |
| **Description** | Every notice will post and update by Cyber center. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 18: Cyber center can announce news and events.**

|  |  |
| --- | --- |
| **FR-18** | Cyber center can announce news and events |
| **Description** | Every news and events will post and update by Cyber center. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 19: Any Research update will be announced by Cyber center.**

|  |  |
| --- | --- |
| **FR-19** | Any Research update will be announced by Cyber center |
| **Description** | Students and teacher’s research paper will be uploaded in this section by Cyber center. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

**Table 20: Cyber center can handle miscellaneous section.**

|  |  |
| --- | --- |
| **FR-20** | Cyber center can handle miscellaneous section |
| **Description** | Only Cyber center can update the emergency issue, officers association, teachers associations. |
| **Stakeholders** | Users, Cyber center |
| **Priority** | High |

#### Data Requirements

Data requirements for this web application project are essential for storing, managing, and retrieving various types of information. The following data requirements are identified:

* NSTU related data.

#### Performance Requirements

Performance Requirements for this web application project are crucial to ensure its smooth operation and responsiveness. The following performance requirements have been identified:

##### Speed And Latency Requirements

The speed and latency requirements for the web application project "NSTU website with integrated chatbot" are as follows:

* + - * Fast Response Time: The Chatbot should provide a quick response to users. The response time should be within a few seconds to ensure a seamless user experience.

Here table 3.3.1.1 shows the speed and latency requirement that is user authentication and authentication with minimal response time

|  |  |
| --- | --- |
| SLR-01 | User Authentication and Authorization with Minimal Response Time |
| Description | The authentication and authorization processes should be efficient and have  minimal response time to enable quick and secure access to the application's features and functionalities. |
| Stakeholders | All users |
| Priority | High |

*Table-3.3.1.1: User Authentication and Authorization with Minimal Response Time*

##### Precision & Accuracy Requirements

The web application should adhere to certain precision and accuracy standards to ensure reliable and accurate results. The following requirements should be met:

* **Response Accuracy**: The chatbot should provide accurate and correct responses to user queries and requests. It should strive to understand the user's intent accurately and generate relevant and precise answers. The accuracy can be measured by evaluating the correctness of the responses against expected or predefined answers.
* **Error Rate:** The chatbot should aim to minimize errors in its responses. The error rate can be measured by tracking the frequency of incorrect or misleading responses provided by the chatbot. The requirement might specify an acceptable maximum error rate that the chatbot should not exceed.
* **Intent Recognition Accuracy:** The chatbot should accurately identify the user's intent or purpose behind their inputs. It should be able to correctly interpret and understand the user's queries, even when they are phrased differently or contain variations. The intent recognition accuracy can be measured by comparing the chatbot's recognized intent against the intended user intent.
* **Continuous Improvement:** The chatbot should have mechanisms in place to continuously learn and improve its accuracy over time. This can involve analyzing user interactions, gathering feedback, and leveraging machine learning techniques to enhance its performance. The requirement might include provisions for regular model updates or retraining cycles to improve accuracy.

##### Capacity Requirements

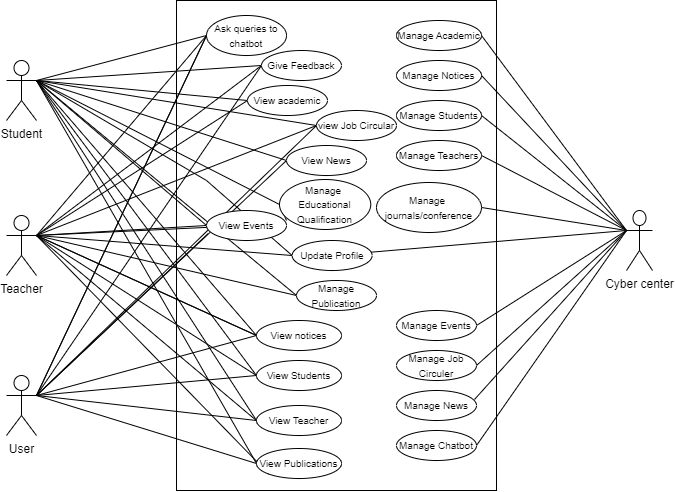
Capacity requirements for a chatbot refer to the system's ability to handle and support a certain volume of user interactions and workload. The capacity requirements for the web application project " **NSTU website with integrated chatbot** " are as follows:

* + - * Scalability: The application should be designed to handle a growing number of users, projects, and data over time. It should have the ability to scale both vertically (increasing server resources) and horizontally (distributing the workload across multiple servers) to accommodate increased usage without compromising performance.
      * Concurrent Users: The system should be capable of supporting multiple concurrent users accessing the application simultaneously. It should be able to handle peak loads during times when a large number of students, supervisors, and project committee members are actively using the system.

# CHAPTER 4 SYSTEM ANALYSIS

#### Use Case Diagram

A use case diagram for a chatbot represents the various interactions and functionalities of the chatbot from a user's perspective. It illustrates the different use cases or tasks that users can perform with the chatbot. The Use Case Diagram for the "NSTU Website with Integrated Chatbot” web application depicts the interactions between Student, Teacher, and Cyber center. Here figure 4.1.1 shows the use case for NSTU Website with Integrated Chatbot.



*Figure – 4.1.1: Use Case Diagram*

#### Use cases descriptions.

A use case description is an in-depth account of the interactions and procedures that take place in a particular scenario or situation where a system, application, or product is used. Here table 01 shows the use case description for Give feedback after received answers, table 02 shows the use case description for Ask queries to the chatbot, table 03 shows the use case description for Manage job circular , table 04 shows the use case description for View Academic Information, table 05 shows the use case description for Manage news and events, table 06 shows the use case description for Manage chatbot , table 07 shows the use case description for Manage Education Qualifications , table 08 shows the use case description for View Teacher Profile and table 09 shows the use case description for View Student Profile.

.

**Table 01: Use Case Description of View Academic Information**

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **01** | |
| **Use case Name** | View Academic Information | |
| **Goal** | View academic successfully | |
| **Preconditions** | Go to NSTU URL | |
| **Success End Condition** | All stakeholders can view academic information successfully | |
| **Failed End Condition** | All stakeholders cannot view academic successfully | |
| **Primary Actors:** | Teacher, Student, Outside user | |
| **Trigger** | Go to NSTU website. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Enter URL http://127.0.0.1:5000/home |
| 2 | Enter Academic Menu |
| 3 | Select Institute/Faculty |
| 4 | View Academic successfully |
| **Quality Requirements** | Applicable | |

## Table 02: Manage Job Circular

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **02** | |
| **Use case Name** | Manage job circular | |
| **Goal** | Cyber center must be managing job circular. | |
| **Preconditions** | Cyber center must be registered to the system. | |
| **Success End Condition** | Cyber center must be managed job circular. | |
| **Failed End Condition** | Cyber center cannot managed job circular. | |
| **Primary Actors:**  **Secondary Actors:** | Cyber center | |
| **Trigger** | Click on ‘admin url’. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Go to admin panel |
| 2 | System checks the information of user. |
| 3 | Click the cyber center menu and upload job circular. |
| 4 | Uploaded successfully |
| **Quality Requirements** | Not applicable | |

## Table 03: Ask queries to the chatbot.

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **03** | |
| **Use case Name** | Ask queries to the chatbot. | |
| **Goal** | All stakeholders can queries using chatbot. | |
| **Preconditions** | Users must be login to the system. | |
| **Success End Condition** | User will successfully ask query in ‘Information system chatbot based on NSTU’. | |
| **Failed End Condition** | User can’t ask query to the chatbot | |
| **Primary Actors:** | Teacher- Students, Outside user and Cyber center | |
| **Trigger** | Click on ‘Chatbot ICON’. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Users click chatbot icon |
| 2 | Search query into text |
| **Quality Requirements** | Applicable | |

## Table 03: Use Case Description of Give Feedback

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **03** | |
| **Use case Name** | Give feedback after received answers | |
| **Goal** | User can provide feedback once system provides the necessary response to the requested inquiry. | |
| **Preconditions** | Users must be login to the system. | |
| **Success End Condition** | User can provide feedback | |
| **Failed End Condition** | User can’t provide feedback | |
| **Primary Actors:** | Teacher- Students, Outside user and Cyber center | |
| **Trigger** | Click on ‘Chatbot ICON’. | |
| **Main Success Flows** | **Step** | **Actions** |
| 1 | Users click chatbot icon |
| 2 | Search query into text |
|  | 3 | System understands the message |
|  | 4 | System replies text message |
|  | 5 | User can provide feedback to this message |
| **Quality Requirements** |  | |

## Table 04: Use Case Description of Manage Student

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **04** | |
| **Use case Name** | Manage Student | |
| **Goal** | Cyber center can manage student information. | |
| **Preconditions** | Cyber center must be login to the system. | |
| **Success End Condition** | Cyber center can create student information. | |
| **Failed End Condition** | Cyber center cannot create student information. | |
| **Primary Actors:** | Cyber center | |
| **Trigger** | Click on ‘ADMIN URL. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Login to the cyber center section |
| 2 | Select the Student menu |
| 3 | Add student information |
| 4 | Uploaded successfully. |
| **Quality Requirements** | Applicable | |

## Table 05: Use Case Description of manage News and Events

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **05** | |
| **Use case Name** | Manage news and events | |
| **Goal** | Cyber center can upload or delete events and news. | |
| **Preconditions** | Cyber center must be login to the system. | |
| **Success End Condition** | Cyber center can be upload or delete events and news. | |
| **Failed End Condition** | Cyber center cannot upload or delete events and news. | |
| **Primary Actors:** | Cyber center | |
| **Trigger** | Click on ‘ADMIN URL. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Login to the cyber center section |
| 2 | Select the News and Event Menu |
| 3 | Add News and Events |
| 4 | Uploaded successfully. |
| **Quality Requirements** | Applicable | |

## Table 06: Use Case Description of Manage chatbot.

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **06** | |
| **Use case Name** | Manage chatbot | |
| **Goal** | Cyber center manage chatbot information and training chatbot. | |
| **Preconditions** | Cyber center must be login to the system. | |
| **Success End Condition** | Cyber manage chatbot information and training chatbot. | |
| **Failed End Condition** | Cyber cannot manage chatbot information and training chatbot. | |
| **Trigger** | Click on ‘ADMIN URL. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Login to the cyber center section |
| 2 | Select the Chatbot menu |
| 3 | See the chatbot response and feedback |
| 4 | Manage successfully. |
| **Alternative Flows** |  | N/A |
| **Quality Requirements** | Applicable | |

## Table 07: Use Case Description of Manage Education Qualifications

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **07** | |
| **Use case Name** | Manage Education Qualifications | |
| **Goal** | Teacher and student can manage own education qualifications | |
| **Preconditions** | Teacher or Student must be login to the system. | |
| **Success End Condition** | Teacher and student can manage own education qualifications | |
| **Failed End Condition** | Teacher and student cannot manage own education qualifications | |
| **Primary Actors:**  **Secondary Actors:** | Teacher, Student | |
| **Trigger** | Go to teacher dashboard | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Login to the teacher dashboard . |
| 2 | Update Educational qualifications |
| 3 | Manage successfully |
| **Alternative Flows** |  | N/A |
| **Quality Requirements** | Applicable | |

## 

## Table 08: Use Case Description of view teacher profile

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **08** | |
| **Use case Name** | View Teacher Profile | |
| **Goal** | All stakeholders can view teacher profile | |
| **Preconditions** | Go to teacher url | |
| **Success End Condition** | All stakeholders can view teacher profile | |
| **Failed End Condition** | All stakeholders cannot view teacher profile | |
| **Primary Actors:**  **Secondary Actors:** | Teacher, Student, Cyber center, Outside user. | |
| **Trigger** | Click on ‘NSTU URL’. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Go to NSTU URL |
| 2 | Click Academic menu and go to faculty |
| 3 | View teacher information |
| 4 | View successfully |
| **Quality Requirements** | Applicable | |

**Table 10: Use Case Description of Reply the query**

|  |  |  |
| --- | --- | --- |
| **Use case Id** | **10** | |
| **Use case Name** | View Student Profile | |
| **Goal** | All stakeholders can view Student profile | |
| **Preconditions** | Go to teacher url | |
| **Success End Condition** | All stakeholders can view Student profile | |
| **Failed End Condition** | All stakeholders cannot view Student profile | |
| **Primary Actors:**  **Secondary Actors:** | Teacher, Student, Cyber center, Outside user. | |
| **Trigger** | Click on ‘NSTU URL’. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Go to NSTU URL |
| 2 | Click Academic menu and go to faculty |
| 3 | View Student information |
| 4 | View successfully |
| **Quality Requirements** | Applicable | |

# 

# CHAPTER 5

**SYSTEM DESIGN AND ARCHITECTURE**

#### System Architecture Overview

I developed the web application "NSTU website with integrated Chatbot" using machine learning(NLP) and Flask.

5.1.1. Executing the application

The chatbot application is built using Flask, a web framework in Python. The application allows users to interact with a chatbot by entering text messages and voice commands. The chatbot utilizes a pre-trained model and a dataset of intents to understand and respond to user queries.

5.1.2. Loading Model and Data: The code loads a pre-trained chatbot model (saved as 'model.h5') and necessary data files (texts.pkl and labels.pkl) for prediction.

5.1.3. Library Function: The code downloads popular NLTK resources and initializes the WordNet lemmatizer for text preprocessing.

5.1.4. Sentence Cleaning: The clean\_up\_sentence function tokenizes and lemmatizes the user's request to prepare it for further processing.

5.1.5. Bag-of-Words (B W) Indexing: The bow function creates a bag-of-words representation of the user's request based on the vocabulary generated from the training data.

5.1.6. Predicting Class: The predict\_class function predicts the class or intent of the user's request using the loaded model. It filters out predictions below a threshold and returns the top predicted intents along with their probabilities.

5.1.7. Language Detection: The code includes functions (is\_english and is\_bangla) to detect the language of the chatbot's response.

5.1.8. Voice Generation: The bangla\_voice and english\_voice functions use the gTTS package to generate audio responses in Bengali and English, respectively. The audio is saved as an MP3 file and played using the Pygame library.

5.1.9. Getting Chatbot Response: The getResponse function retrieves a suitable response from the intents data based on the predicted intent class.

5.1.10. Routes and Templates: The code defines different routes for various webpages, such as the index page, user registration, user login, chatbot response retrieval, user feedback submission, user logout, Cyber center login, Cyber center dashboard, adding queries based on feedback, removing feedback, and Cyber center logout. Corresponding HTML templates are rendered for these routes.

#### Training the model’s brain

Training the data refers to the process of teaching a machine learning model to understand and recognize patterns within a given dataset. In the context of the provided JSON file containing intents for a chatbot, training the data involves preparing the model to understand user queries and respond accordingly.

5.2.1. Reading Dataset: A json file called data.json stores all the dataset used to train the model. There are four keys - tag, patterns, responses, context.

5.2.2. Tokenizing and preprocessing: It tokenize the patterns from the JSON file and performs preprocessing tasks such as lemmatizing, removing duplicates, and removing stop words.

5.2.3. Displaying progress and saving serialized data: It displays the number of documents, classes, and unique lemmatized words. It then saves the preprocessed data (words and classes) as serialized pickle files.

5.2.4. Creating training data: It creates training data by indexing tokens. It iterates over the documents and creates a bag of word representation for each pattern, associating it with the corresponding class.

5.2.5. Randomly organizing training data: The training data is shuffled randomly to avoid any bias and is converted into a NumPy array.

5.2.6. Creating the neural model: It defines a sequential neural network model using Keras. It consists of three layers: two hidden layers with dropout regularization and a final output layer with a softmax activation function.

5.2.7. Compiling and training the model: The model is compiled using the stochastic gradient descent (SGD) optimizer and categorical cross-entropy loss function. It is then trained on the training data for a specified number of epochs.

5.2.8. Saving the model: After training, the model is saved as an HDF5 file.

#### Detailed Design and Component Selection

Here is a high-level overview of the Detailed Design and Component Selection:

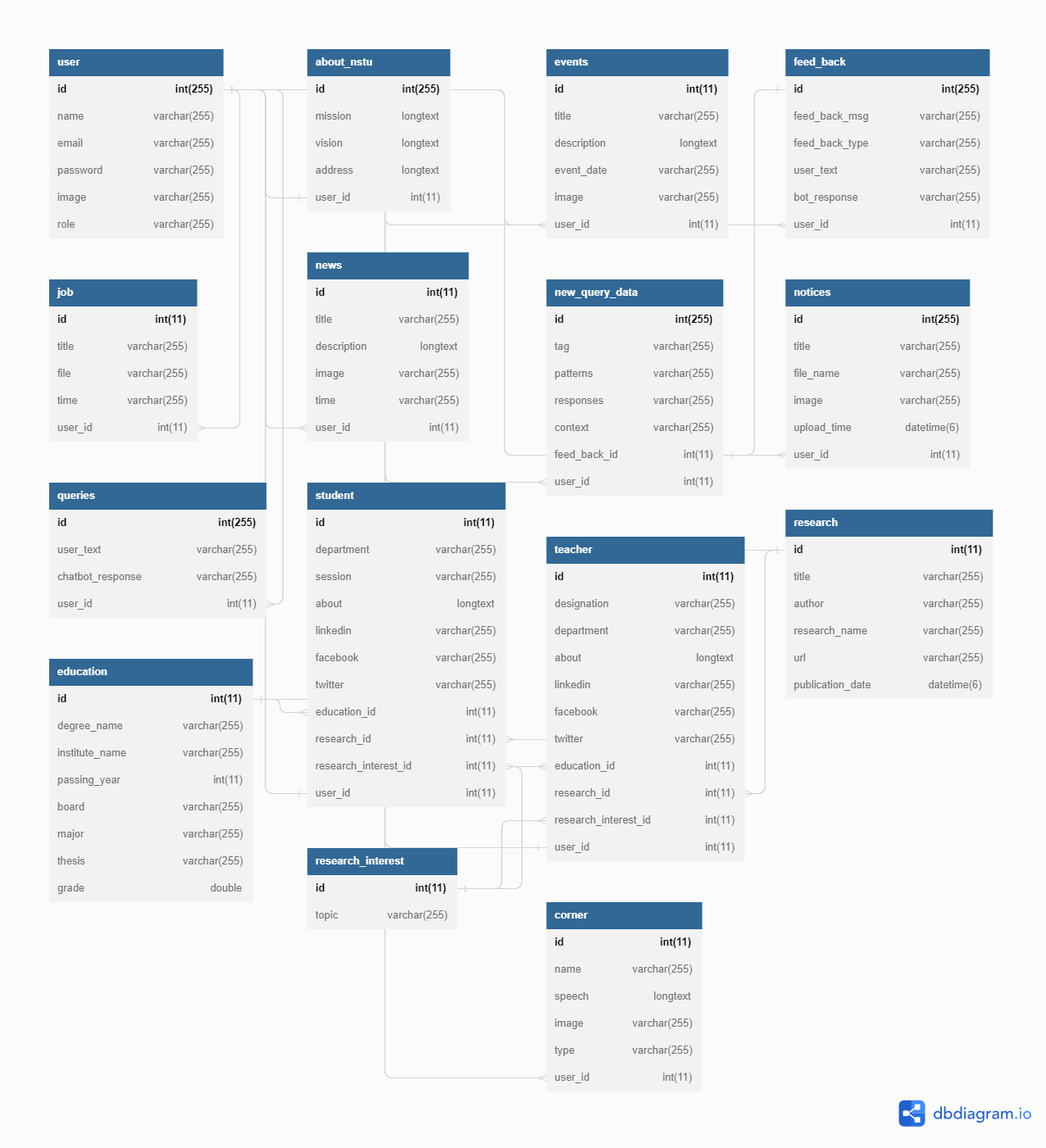
##### Front-end Development:

* + - * I utilized HTML, CSS, bootstrap and MUI to create the visual elements, layout, and interactive components.
      * I implemented responsive design techniques to ensure the application is accessible on various devices and screen sizes.

##### Back-end Development:

* + - * I used MySQL database and python web framework(FLASK)

#### Database Schema



#### User Interface Design (Techniques & Strategies)

##### Responsive Design:

* + - * I employed responsive design techniques to ensure that the web application is accessible and provides an optimal user experience across various devices and screen sizes, including desktops, laptops, tablets, and mobile phones.
      * I utilized CSS media queries to adjust the layout, typography, and elements' positioning based on the screen resolution.

##### Clear and Organized Layout:

* + - * I designed the user interface with a clean and organized layout to avoid clutter and confusion.
      * I used grids and columns to structure the content and maintain a logical flow of information.
      * I employed visual hierarchy techniques, such as font size, color, and contrast, to highlight important elements and guide users' attention.

##### Interactive Elements:

* + - * I incorporated interactive elements, such as buttons, forms, dropdowns, and tooltips, to enhance user engagement and provide intuitive interactions.
      * I employed feedback mechanisms, such as hover effects and button states, to provide visual cues and confirm user actions.

##### Visual Design and Branding:

* + - * I followed a visually appealing design approach that aligns with the project's branding and target audience.
      * I utilized a consistent color scheme, typography, and visual elements, such as icons and illustrations, to create a cohesive and memorable visual identity.
      * I used inline validation to provide real-time feedback to users while filling out forms, helping them correct mistakes before submitting.

##### Usability Testing and Iteration:

* + - * I conducted usability testing sessions with target users to gather feedback on the user interface design and identify areas for improvement.
      * Based on the feedback received, I iterated on the design, making necessary adjustments to enhance usability and address any usability issues

# CHAPTER 6 IMPLEMENTATION AND DEVELOPMENT

#### Development Tools & Technology

During the development of the web application project ”**NSTU Website with integrated Chatbot**”, I used a range of tools and technologies including PyCharm, HTML5, CSS3, Bootstrap, Font Awesome, MUI CSS, Speech Recognition, gTTs, Lang Detect  with Flask, MySQL, NLP(NLTK), Tensorflow, Keras

##### User Interface Technology

For **NSTU Website with integrated Chatbot** project I utilized popular development tools and technologies such as PyCharm, HTML5, CSS3, Bootstrap, FontAwesome, MUI CSS, Flask, MySQL. These tools and technologies provided a robust foundation for building the application's frontend and backend components, ensuring efficient development and seamless functionality.

##### HTML5

In the development of the " **NSTU website with integrated chatbot** " web application, I opted to use HTML as the markup language.HTML is the standard markup language used for creating the structure and content of web pages. It defines the elements, tags, and attributes used to structure and present information on the World Wide Web. It is the backbone of the World Wide Web and is interpreted by web browsers to display web content.

##### CSS3

For **NSTU Website with integrated Chatbot** I used CSS for user interface designing. It is the standard stylesheet language used to define the visual appearance of HTML documents. It allows me to control the layout, colors, fonts, spacing, and other visual aspects of web pages.

##### Bootstrap and MUI CSS

**For NSTU Website with integrated Chatbot** I incorporated Bootstrap & MUI CSS as the CSS framework. Bootstrap is a widely used open-source CSS framework which provides a collection of CSS components, styles, and layout grids that can be easily incorporated into web projects. Bootstrap helps me create consistent and responsive designs, as it offers a mobile-first approach and includes a responsive grid system that adapts to different screen sizes. MUI CSS is a CSS framework that implements Material Design. It emphasizes theming and customization allowing me to create unique and branded designs.

##### Font Awesome

Font Awesome was utilized in the " **NSTU Website with integrated Chatbot** " web application to enhance the visual appeal and user experience by incorporating a wide range of scalable vector icons. These icons were employed across various components and features to provide intuitive navigation, improve information representation, and enhance overall aesthetics.

##### Speech Recognition

JavaScript's Speech Recognition API enables the integration of speech recognition into web applications. By leveraging this API, I added speech recognition to the " **NSTU Website with integrated Chatbot** “web app, enhancing its usability. Users could interact with the app using voice commands, enabling hands-free operation. The API captured spoken language, converting it to text for application use.

##### Implementation Tools and platform

For the implementation of the " **NSTU Website with integrated Chatbot** " web application, I utilized a range of tools and platforms. I developed the application using PyCharm as the integrated development environment (IDE) and leveraged the Flask(Python web framework), MySQL(Database) for the back-end. The application was hosted on localhost. Additionally, I utilized Git for version control.

##### MySQL

For the " **NSTU Website with integrated Chatbot** " web application, MySQL served as the chosen relational database management system (RDBMS). It played a critical role in storing and managing diverse data pertaining to students, supervisors, project committees, notices, and project progress. MySQL offered a robust and scalable database solution, ensuring the reliability and scalability required for the application's data needs. With its strong SQL support, MySQL facilitated efficient data retrieval and manipulation, enabling seamless integration and manipulation of information. Throughout the development process, MySQL's dependable performance contributed to the smooth operation of the application and aided in the successful implementation of data-related functionalities.

##### Apache HTTP Server

The " **NSTU Website with integrated Chatbot** " web application relied on Apache HTTP Server as its web server. Apache played a crucial role in hosting the application and efficiently handling HTTP requests and responses. Renowned for its power and reliability, Apache provided a robust platform to serve web content to users. Its versatility and extensive feature set made it a fitting choice for delivering the application's web content, ensuring its availability and accessibility. Apache's exceptional performance capabilities further contributed to seamless user experiences. Additionally, Apache's strong security measures, support for various modules, and customizable configurations provided a secure and efficient web hosting environment for the application, safeguarding critical data and maintaining system integrity.

#### Flask

Flask was the chosen web framework for the development of the " **NSTU Website with integrated Chatbot** " web application. With its simplicity and flexibility, Flask provided a powerful foundation for building the application's backend. Its lightweight nature and intuitive design made it easy to understand and work with, allowing for rapid development and iteration. Flask's extensive documentation and active community support were invaluable resources throughout the development process. Leveraging Flask's routing capabilities, request handling, and template rendering, we were able to create dynamic and interactive web pages seamlessly. Additionally, Flask's modular architecture enabled us to easily integrate various extensions and libraries to enhance the application's functionality, such as Flask-Login for user authentication and Flask-SQL Alchemy for seamless database integration. Overall, Flask proved to be a reliable and efficient framework, empowering us to deliver a robust and user-friendly web application.

#### Language Detection

Language detect is a Python library for language detection. I incorporated language detect in the " **NSTU Website with integrated Chatbot** " web app to automatically detect the language of text input. This facilitated language-specific processing and customization, improving multilingual support. Language detects API seamlessly integrated into the application, accurately detecting the language of text strings or documents. With support for numerous languages, language detect enhanced language-related features, providing a personalized experience for users across different languages.

#### gTTS

gTTS (Google Text-to-Speech) is a Python library that converts text to speech using Google's Text-to-Speech engine. In the "SPL 1, 2 & Final Year Project Management System" web app, I used gTTS to add text-to-speech functionality. It transformed text into spoken audio, enhancing accessibility and user experience. gTTS's API made it easy to generate audio files from text in multiple languages, with options to control speed, language, and output format. By incorporating gTTS, the app could provide auditory feedback, read notifications, and deliver dynamic voice-based responses. This integration expanded accessibility and improved the overall user experience of the web application.

#### Machine Learning

Machine learning and NLP were pivotal in developing the " **NSTU Website with integrated Chatbot** " web app. ML models automated tasks like sentiment analysis and text classification. NLP techniques, such as tokenization and part-of-speech tagging, enabled effective language processing. Integration of ML and NLP enhanced the app's functionality with document summarization, language translation, and intelligent chatbot interactions. This improved user experience and productivity. The system's ability to understand human language made it more efficient and user-friendly.

#### NLTK

NLTK (Natural Language Toolkit) played a crucial role in developing the " **NSTU Website with integrated Chatbot** " web app. Leveraging NLTK's linguistic data and algorithms, we achieved advanced natural language processing. Tasks like tokenization, part-of-speech tagging, and named entity recognition were implemented using NLTK. Its modules facilitated sentiment analysis and text classification, enhancing app functionality. NLTK empowered the system to extract insights from textual data and provide intelligent language-based features. It proved instrumental in building a sophisticated language-driven web app.

#### Keras

Keras, a high-level neural networks API, played a critical role in the development of the " **NSTU Website with integrated Chatbot** " web app. With its user-friendly interface and extensive set of tools, Keras simplified the process of building deep learning models. It provided a seamless abstraction layer over low-level frameworks, enabling rapid prototyping and experimentation. Keras supported various neural network architectures and allowed for easy customization. By leveraging Keras, we achieved advanced deep learning capabilities, empowering the web application with powerful and efficient machine learning algorithms.

#### PICKLE

Pickle, a built-in Python module, proved invaluable in the " **NSTU Website with integrated Chatbot** " web app. It offered seamless object serialization and deserialization, enabling easy storage and retrieval of Python objects. Pickle allowed us to save trained models, data structures, and other important objects, preserving their state and structure. This facilitated efficient sharing, persistence, and reusability of objects across different sessions or environments. Pickle's simplicity and compatibility with various Python libraries made it a convenient choice for data persistence in the web application, ensuring the smooth and reliable operation of the system.

#### NUMPY

NumPy, a fundamental library for numerical computations, played a crucial role in the " **NSTU Website with integrated Chatbot** " web app. With its powerful n-dimensional array objects and comprehensive mathematical functions, NumPy facilitated efficient handling and manipulation of numerical data. It provided an essential foundation for various data processing tasks, including array operations, linear algebra, and statistical analysis. NumPy's optimized performance and seamless integration with other libraries made it an indispensable tool for scientific computing. By harnessing NumPy, we ensured efficient and accurate numerical computations within the web application.

#### JSON

JSON (JavaScript Object Notation) was essential in training data for the " **NSTU Website with integrated Chatbot** " web app. It provided a lightweight and readable format for structuring and storing data. JSON's flexibility allowed seamless integration with various programming languages and frameworks. It facilitated data serialization and deserialization, enabling easy storage and retrieval of structured information. With JSON, data was represented in key-value pairs and nested structures, ensuring a standardized format for training datasets, configuration files, and API responses. JSON's simplicity and compatibility contributed to efficient data handling and data-driven functionalities in the web application.

# CHAPTER 7

**TESTING AND QUALITY ASSURANCE**

#### Testing Features

The testing features implemented in the " **NSTU website with integrated chatbot** " web application include:

* + - Unit Testing: I conducted unit tests to verify the functionality of individual components and modules, ensuring they performed as expected.
    - Integration Testing: I performed integration tests to validate the interaction and interoperability of different modules within the application.
    - User Acceptance Testing: I engaged users, including faculty members and students, to participate in user acceptance testing. This involved evaluating the application's usability, functionality, and adherence to requirements.
    - Regression Testing: I conducted regression tests to ensure that new features or bug fixes did not introduce any unintended issues or regressions in the existing functionality.
    - Performance Testing: I assessed the application's performance under various loads and stress levels to identify potential bottlenecks and optimize its responsiveness and scalability.
    - Security Testing: I conducted security tests to identify and address vulnerabilities, ensuring that user data and the application were adequately protected from unauthorized access or malicious attacks.

##### Features to be tested

The features to be tested for the " **NSTU Website with integrated Chatbot** " web application include:

* + - * Student Module
      * Cyber center Module
      * Teacher Module

##### Features not to be tested

The features not to be tested for the " **NSTU website with integrated chatbot**” web application project may include:

* + - * Third-party libraries or frameworks
      * Operating system compatibility
      * Network or server infrastructure
      * External APIs or services

#### Testing Strategies

The testing strategies for the " **NSTU Website with integrated Chatbot** " web application project may include:

* + - Functional Testing: Verify that all features and functionalities of the application are working as expected. This includes testing user registration, login, ask query via text and voice, edit feedback response, receive answer via text and voice.
    - User Interface Testing: Ensure the user interface is intuitive, responsive, and visually appealing. Test the layout, navigation, forms, buttons, and other user interface elements across different devices and screen sizes.
    - Integration Testing: Validate the integration of different modules and components within the application.
    - Performance Testing: Evaluate the performance of the application under expected user load. Test the application's response time, resource usage, and scalability. Identify any performance bottlenecks and optimize the application as needed.
    - Compatibility Testing: Ensure the web application functions correctly across different web browsers, including popular ones like Chrome, Firefox, Safari, and Edge.
    - Usability Testing: Gather feedback from users to assess the application's ease of use, intuitiveness, and overall user satisfaction. Incorporate user feedback to improve the user interface and enhance the user experience.
    - Regression Testing: Perform regression tests after making changes or adding new features to ensure that existing functionalities are not adversely affected.
    - Data Integrity Testing: Verify that data is stored, retrieved, and manipulated correctly within the application's database.
    - Accessibility Testing: Assess the accessibility of the web application for users with disabilities. Test for compliance with accessibility standards and guidelines, such as WCAG (Web Content Accessibility Guidelines).

##### Test Approach

The test approach I followed consisted of the following steps:

* + - * Requirement Analysis: I reviewed the project requirements and specifications to gain a thorough understanding of the desired functionalities and features.
      * Test Planning: I created a comprehensive test plan outlining the testing objectives, scope, resources, timelines, and deliverables. I identified the different types of testing to be performed and prioritized them based on criticality.
      * Test Design: I developed detailed test scenarios, test cases, and test scripts for each identified feature and functionality. I considered both positive and negative test scenarios to ensure complete test coverage.
      * Test Environment Setup: I set up the necessary test environment, including the required hardware, software, databases, and network configurations. I configured test data and ensured the environment closely resembled the production environment.
      * Test Execution: I executed the test cases according to the planned test scenarios. I recorded the test results, including any defects or issues encountered during testing.
      * Test Reporting: I generated test reports summarizing the testing activities, test coverage, defects found, and their status. I provided clear and concise documentation of the test results for stakeholders and project management.
      * Test Completion and Sign-off: I conducted test completion activities, including regression testing, to ensure that all defects had been addressed and the application met the defined quality standards.
      * Test Documentation: I maintained proper documentation of test artifacts, including test plans, test cases, test scripts, and test results. This documentation served as a reference for future maintenance and testing efforts.
      * Continuous Improvement: I continuously evaluated and refined the testing process based on lessons learned and feedback received. I identified areas for improvement and implemented necessary changes to enhance the overall test approach and efficiency.

##### Pass/Fail Criteria

During testing, I employed the following pass/fail criteria to determine the success or failure of the test cases:

* + - * Functional Requirements: A test case is considered a pass if the functionality being tested behaves as expected and meets the defined requirements. It is considered a fail if the functionality does not perform as intended or deviates from the specified requirements.
      * Usability: A test case is deemed a pass if the user interface is intuitive, user-friendly, and provides a seamless user experience. It is considered a fail if the interface is confusing, difficult to navigate, or presents usability issues that hinder user interaction.
      * Performance: Performance-related test cases are evaluated against predefined benchmarks and thresholds. If the application performs within acceptable response times, handles concurrent user loads efficiently, and does not experience any performance degradation, the test case is marked as a pass. It is considered a fail if the performance falls below the defined standards or results in excessive delays or errors.
      * Compatibility: Test cases related to compatibility involve verifying the application's compatibility with different browsers, devices, and operating systems. A test case is considered a pass if the application functions correctly across the specified platforms. It fails if compatibility issues are identified, resulting in improper rendering or functionality on certain configurations.
      * Integration: Test cases for integration assess the seamless interaction between different modules, components, or third-party systems. A test case passes if the integration is successful, with data flowing correctly and communication between systems functioning as expected. It fails if integration issues, such as data loss, incorrect data transfer, or system crashes, are encountered.

### Test Case Execution

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case**  **Scenario** | **Test Steps** | **Test Data** | **Expected**  **Results** | **Actual**  **Result** | **Pass/Fail** |
| TC-1 | chatbot should be able to reply to user inquiries effectively | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.User write message  5.Chatbot replies to the message | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234”  User message: “IIT”  Chatbot replies:  “Institute of Information Technology” | Chatbot replies to the user’s inquiries | As Expected, | Pass |
| TC-2 | Chatbot should handle ambiguous queries | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.User write message  5.Chatbot replies to write correct message | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234”  User message: “IIT”  Chatbot replies:  “please write relevant message” | Chatbot handles user’s ambiguous queries | As Expected, | Pass |
| TC-3 | Users may text their inquiries. | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.User write text message | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234”  User message: “NSTU” | Users text their inquiries. | As Expected, | Pass |
| TC-4 | Users can ask query via voice | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.User ask voice message | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234”  voice message: “Tell me about NSTU” | Users deliver voice message successfully | As Expected, | Pass |
| TC-5 | Users must receive text messages as responses. | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.User write text message  5.User receive text responds from the system | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234”  User message: “IIT”  Chatbot replies:  “Institute of Information Technology” | text messages as responses successfully | As Expected, | Pass |
| TC-6 | Users must be receiving answer via voice | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.User write text message  5.User receive voice message | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234”  User message: “IIT”  Chatbot replies:  “Institute of Information Technology” | Users receiving voice answer | As Expected, | Pass |
| TC-7 | Users can login to the system | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.Clik on Login  4.User write mail and password | User name: “[adnansami@gmail.com](mailto:adnansami@gmail.com)”  Password: “1234” | Login successfully | As Expected, | Pass |
| TC-8 | Users can register to the system | 1.Go to localhost  “http://127.0.0.1:5000/”  2.User login to the system  3.User click on chatIcon  4.Clik on Registration  5.User write mail and password | User name: “[adnan@gmail.com](mailto:adnan@gmail.com)”  Password: “12345” | registration successfully | As Expected, | Pass |
| TC-9 | Cyber center must be editing feedback information. | 1.Go to localhost  “http://127.0.0.1:5000/Cyber center”  2.click edit option |  | editing feedback information successfully | As Expected, | Pass |
| TC-10 | Cyber center must be storing feedback data | 1.Go to localhost  “http://127.0.0.1:5000/Cyber center”  2.click edit option  3.write response | Message: “Noakhali university of science and technology”  Response: “Noakhali Science and Technology University” | Storing info successfully | As Expected, | Pass |
| TC-11 | Cyber center manage teacher information | 1.Go to localhost  “http://127.0.0.1:5000/Cyber center”  2.click teacher  3.fillup information |  |  | As Expected, | Pass |

##### Traceability Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements Traceability Matrix** | | | | | | | |
| **Project Name** | NSTU website with integrated chatbot | **Business Area** | | N/A | | | |
| **Project Leader** | Md Al Adnan | **Business Analyst Lead** | | Md Al Adnan | | | |
| **QA Lead** | Md Al Adnan | **Target Implementation Date** | |  | | | |
| **Category/ Functional Activity** | **Requirement Description** | **Use Case Reference** | **Design Document Reference** | **Code Module/ Reference** | **Test Case Reference** | **User Acceptance Validation** | **Comments** |
| **FR-01** | The chatbot should be able to reply to user inquiries effectively | **UC-01** |  |  |  | **Pass** |  |
| **FR-02** | The Chatbot should handle ambiguous queries | **UC-02** |  |  | **TC-02** | **Pass** |  |
| **FR-03** | Users may text their inquiries. | **UC-04** |  |  |  | **Pass** |  |
| **FR-04** | Users can ask query via voice | **UC-05** |  |  | **TC-04** | **Pass** |  |
| **FR-05** | Users must receive text messages as responses. | **UC-06** |  |  | **TC-05** | **Pass** |  |
| **FR-06** | Users must be receiving answer via voice | **UC-07** |  |  | **TC-06** | **Pass** |  |
| **FR-07** | Users can log in to the system | **UC-08** |  |  |  | **Verified** |  |
| **FR-08** | Users can register to the system | **UC-15** |  |  | **TC-08** | **Pass** |  |
| **FR-09** | Cyber center must be editing feedback information. | **UC-13** |  |  | **TC-11** | **Verified** |  |
| **FR-10** | Cyber center must be storing feedback data | **UC-09** |  |  |  |  |  |

*Table – 7.2.5.1: Requirements Traceability Matrix Table*

#### Testing Environment (hardware/software requirements)

During the manual testing process of the web application project, I utilized the Chrome Developer Tools extensively, along with console logging for debugging and testing purposes. The testing environment for manual testing had the following hardware and software requirements:

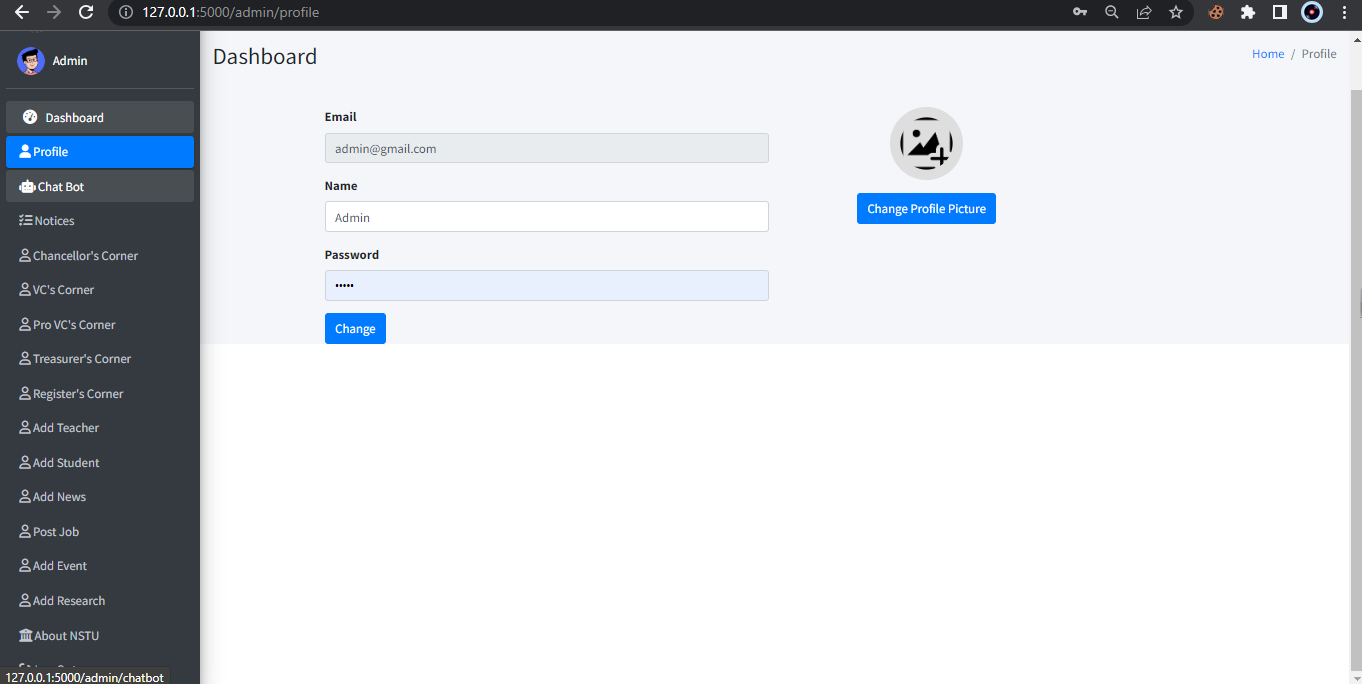
* + - Hardware Requirements: I needed a computer system capable of running the web application and supporting the required software components. This typically included a computer with sufficient processing power, memory, and storage capacity to accommodate the web application and testing tools.
    - Software Requirements:
      * Operating System: I used an operating system compatible with the web application and the selected development tools. This could be Windows, macOS, or Linux.
      * Web Browser: I primarily relied on the Google Chrome web browser, as it provides comprehensive developer tools, including the Chrome DevTools for inspecting and debugging web applications.
      * Text Editor: I utilized a text editor like Visual Studio Code or Sublime Text for reviewing and modifying the application's source code during testing.
      * Console Logging: I employed console logging statements within the JavaScript code to track and analyze the application's behavior, debug issues, and verify expected outputs.
      * Network Connectivity: A stable internet connection was necessary to access the web application, simulate various network conditions, and test its responsiveness.

# CHAPTER 8 USER MANUAL

#### User Manual (Cyber Center)

**User Manual: Cyber center**

Cyber center can manage chatbot, Notice, About, Administration corner, Academic Corner(Add teacher, add student), News, Events, Job, Journals.



#### User Manual (Teacher)

Teacher can update their own information and also view all information provide by cyber center also asked queries using chatbot.

#### A screenshot of a computer Description automatically generated

#### 

#### User Manual (Student)

Student also can update their own information and also view all information provide by cyber center also asked queries using chatbot.

A screenshot of a computer

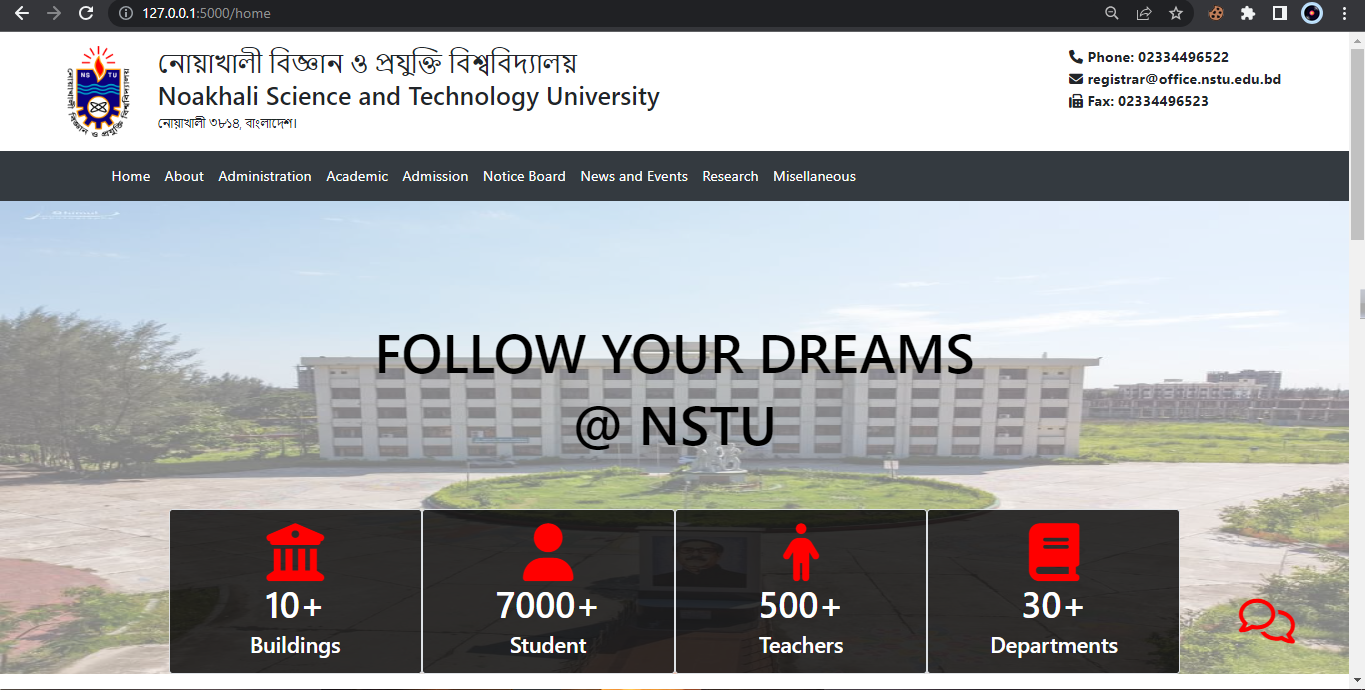
Description automatically generated

A person in a black shirt

Description automatically generated

#### User Manual (Outside User)

Outside user means they are not directly related to NSTU, they can be guidance or any other person. They can use view information provided by cyber center and also can use chatbot.



A screenshot of a chatbot

Description automatically generated

# CHAPTER 9 PROJECT SUMMARY

#### Github Link

The GitHub repository link for this web application project is available for access and collaboration.

Link - *https://github.com/aladnansami/NSTU-Website-with-Integrated-Chatbot.git*

#### Critical Evolution

Critical Evolution for this web application project involved continuous monitoring, feedback gathering, and iterative enhancements to ensure its optimal performance, usability, and reliability. Throughout the development process, I conducted thorough testing, identified and resolved bugs, and incorporated user feedback to improve the overall functionality and user experience. I also stayed updated with the latest technologies and industry trends to address any potential security vulnerabilities and ensure the application remains robust and future-proof.

#### Limitations

The project has a few limitations due to incomplete features. These limitations are as follows:

* + - Cannot manage office information.
    - Cannot manage staff information
    - Lack of Huge data

#### Future Scope

The completion of the web application project opens up several possibilities for future enhancements and expansions. Here are the potential areas of future scope for this project:

* + - Multi-Modal and Multi-Channel Support
    - Cognitive Chatbots and Emotional Intelligence
    - Integration with IoT
    - Advanced Natural Language Processing (NLP)
    - Build Fully dynamic NSTU website will a lot of functionalities.
    - Fully Dynamic Management For NSTU.

**References**

1. D. P. J. K. A. R. Brian Birnbach, Software & Systems Requirements Engineering: In Practice, 1st ed., McGraw Hill.
2. K. W. a. J. Beatty, Software Requirements, Redmond, Washington 98052-6399: Microsoft Press, 2013.