WEB DEVELOPMENT FOR DIFFERENT INTERESTING SYSTEMS

Student Name: Rasagya Shokeen
Roll Number: 2019088

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering
on May 02, 2022

BTP: Engineering Track

BTP Advisor
Dr. Rajiv Ratn Shah

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **W**EB DEVELOPMENT FOR DIFFERENT INTERESTING SYSTEMS submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr. Rajiv Ratn Shah**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

Rasagya Shokeen **Place & Date: IIITD, 02/05/22**

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Rajiv Ratn Shah **Place & Date: IIITD, 02/05/22**

Abstract

My B.Tech project is based on Web development, wherein I was supposed to build a website and deploy an NLP model to implement text processing and generate reports based on the interpretation of the processed text. The complete website would be a fully functional website with multiple users and stakeholders divided broadly in three categories; the students, the instructors and the admin. Each of the designations have their own functionalities which have been explained in detail below. The entire project is made to help underprivileged students/adults who do not have access to education/educational institutions or do not have the monetary means to afford education or attain literacy. This platform is meant to help them attain linguistic abilities presently in English(could be expanded to other languages) with the click of a few buttons just with their smartphones and popular social networking platforms such as WhatsApp which can be commonly found in remote locations as well. This project has been developed in association with i-Saksham which is an NGO based in Chattisgarh which aspires and has been teaching students and underprivileged people in Chattisgarh the art of communication using English as a medium of communication.

Keywords: Natural Language Processing, Web development, Designations, Underprivileged, i-Saksham.

Acknowledgments

It would not have been possible to work on this project without the support and guidance of many people whose names may not all be enumerated. Their contributions are sincerely acknowledged and appreciated. I would however like to express my deepest gratitude and appreciation to the following people:
Dr. Rajiv Ratn Shah for providing me with the opportunity to work on this project, and for his support, guidance and encouragement during the course of this project. To my institute, Indraprastha Institute of Information Technology (IIIT Delhi) for providing me with the platform to experiment and explore with such a project. To all the family and friends who provided moral and emotional support during the course of this project.
I thank you all.

Work Distribution

The first phase of the BTP focussed on the web application which used Javascript, HTML, CSS, Bootstrap as the frontend stacks and Flask as the backend framework. With SQLite serving as the database for the website.
The very first task was to develop the NLP model to perform text processing, index slicing, extracting etc. This was implemented with the help of inbuilt libraries as well as imported libraries of python NLTK package.
The next task was to expand the report generation from english language to hindi language. This required the developer to use a different set of libraries, parse the hindi language and create different functions which were to be called in case the user selected hindi as the language in which the report is to be generated.
The next task was to develop the application, which was completed in two parts; the frontend,

for which Bootstrap, JavaScript, HTML, CSS were used as technologies to create a user friendly frontend of the application. The backend, Flask was used to create the backend of the website which would handle the authentication as well the NLP models and functions which were to be executed on the input received from the frontend.

The second phase of the BTP focussed on the more advanced web application which would be built using the stack: ReactJS, Axios, Redux, Django, Django-rest-framework. Herein the application was expanded to multiple users as 3 stakeholders; the students, the instructors and the admin.

The first task was to finalize the stacks which would be the most appropriate for the creation of the website.

Then the student pages were designed and constructed, post which the instructor pages were created and then the admin pages were created. This required extensive modeling and planning as to what would be the flow of the website, how would the stakeholders communicate with each other and what would be the general layout of the website. The details of the above are presented in the lower sections of this document.

Contents

1 Introduction **1**

2 Architecture, design, implementation, and validation of the solution **2**

2.1 HTML, Bootstrap, Flask model: 2

2.2 Natural Language Processing model working: 2

2.3 Design: 3

2.3.1 Advanced website design: 3

2.4 ReactJs, Django portal: 4

3 User Manual **5**

3.1 For administrators: 5

3.2 For instructors: 5

3.3 For students: 6

4 Layout of the web application **8**

5 Conclusion and Future Work **16**

Chapter 1

Introduction

Currently a major problem that we can observe in society is the lack of a scalable platform to teach the underprivileged people basic languages. A similar problem was identified by an NGO named i-saksham for the underprivileged people that lived in Chhattisgarh. In these areas, it is practically difficult for teachers to physically go and teach these people hence they decided to teach them basic subjects via an online mode using primitive technologies like Whatsapp and google meet. In order to ease the whole process and make a wider impact we are trying to automate the process of evaluation of the assignments that these people do and share over Whatsapp groups via text messages. This had to be done as the number of teachers are few while the audience that needs education is quite large in these areas. Technologies like grammarly and paid resources are not feasible as they are quite expensive for NGOs to afford. Moreover, the people in these areas have very little knowhow of technologies and hence a solution which used only primitive technologies was required so that it doesn't force a major technology in their life which could have demotivated them. The audience includes underprivileged children, women and even men from naxal affected areas in Chhattisgarh. There are people of higher age groups also who are eagerly willing to learn these subjects. The frontend will be managed by the mentors who will provide the basic training to these children and the rest of the audience.

Chapter 2

Architecture, design, implementation, and validation of the solution

There have been two implementations of the web application.

2.1 HTML, Bootstrap, Flask model:

The first one, using SQLite, Flask, Bootstrap, HTML, CSS has been completed with some minor UI based modifications to be made. The application consists of an Authentication system which is being backed by SQLite which is used to store the data of the user. FLask provides us with an effective functionality of authenticating the user before allowing them to view the hidden pages such as the one where the user is supposed to upload the details as well as the exported whatsapp chat .txt file post signing in. If the user tries to directly access the upload page, he/she will be redirected to the signin page maintaining complete content control on the application. Post authentication, the uploaded text file, the dates selected by the user, the whatsApp file details i.e. are the essays/texts written by multiple students or a single student, and finally the language in which the report must be generated. The backend of the code is supported by a Natural language Processing(NLP) model written in python language.

2.2 Natural Language Processing model working:

It works in the following way; it accepts the text file from the frontend through the flask framework then processes it using various functions, extracting the name of the student(s) by using the strip method provided in python language, the essays typed by the students are also

extracted in a similar way, now by using pre-existing NLP libraries as well as standard python libraries such as Numpy, Pandas, datetime, matplotlib etc. the extracted essays are evaluated and analysed for the errors(grammatical as well as punctuation) that have been committed by the students. All the essays are evaluated and analysed during the running of the code and then a report is generated in a pdf format for the administrator/student. At the moment, the report is stored in the folder containing the entire project. It could be further expanded to sending the report by email, message, WhatsApp or even displaying the output on the Web application. The report is generated using the Fpdf library provided by python as various plots, such as a pie chart for the fraction of the type of errors committed, a pie chart to show the number of assignments completed to the number of assignments not attempted etc. Additionally, a detailed report is also provided on the following pages, focussing on individual sentences and in depth analysis and reporting of errors.

2.3 Design:

There has been a dedicated effort to make the report as reader friendly as possible as suggested by the Head of the NGO i-Saksham, greeting lines in the beginning of the report, encouraging phrases on good performance and motivating quotes on below par performance has been provided.
Generating the report in a different language(Hindi): To accommodate a wider section of the society, it required the developer to have the report generated in Hindi language as well, as the one provided in English might prove to be difficult to comprehend for the underprivileged as it is assumed that they have some challenges in understanding basic English language. And hence, an option to generate the report in hindi was added. The user can check a checkbox choosing between the language of the report that he/she wishes to generate. The backend code would see this value as a binary and call the appropriate functions according to the requirement.

2.3.1 Advanced website design:

The website has been designed by using the following: **For styling:** React-Bootstrap libraries which provide user friendly components, such as bootstrap cards, navigation bars, menus, buttons etc. Hamburger menus have also been created to display the profile of the user which appears on the screen on clicking the button on the top right. Cards to display the student details with interactive buttons have been used to make the website user friendly.

For communication: Axios, this has been used to communicate with the server of the website i.e. with the Django-rest-framework. Axios is promise-based, which gives you the ability to take advantage of JavaScript's async and await for more readable asynchronous code. You can also intercept and cancel requests, and there's built-in client-side protection against

cross-site request forgery.

For state management: Redux has been used to control the state of the website in real time, to store local values and instructions. "React Redux is the official React binding for Redux. It allows React components to read data from a Redux Store, and dispatch Actions to the Store to update data. Redux helps apps to scale by providing a sensible way to manage state through a unidirectional data flow model."

For the backend: Django framework will be used for the backend of the website to provide server support to the application. Several dependencies of Django will be used such as Django-rest-framework, Pillow etc.

2.4 ReactJs, Django portal:

The final application would be deployed using the ReactJS and Django stacks using react-bootstrap, Axios etc. The application has been designed using the latest version of ReactJS which is "a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies." The frontend of the application has been designed as well as developed using ReactJS and react-bootstrap. Implementing different functionalities such as Hamburger menus, Navbars, etc. Authentication will be added to the application to Sign Up to the Website, and login while accessing the features. Additional features such as signing up using google accounts can be added as well. The frontend consists of the login page, the signup page, the hamburger menus as well as the upload page to upload the exported files and other details. The UI has been designed to make it as pleasing to the eye as possible and as user friendly as possible giving out immediate feedback and response. After tuning the code for the model, I started developing the web application, which consists of simple web pages with a form field to take inputs for the model. After processing the information, I wanted to render a new web page that shows us the result and stores the user's report in the database. I created the base python file (app.py) that contains the main code of the model as well as the functions that send GET and POST requests for all the pages. The Python interpreter executes this main code to run the Flask web application. I then created the HTML pages for login, signup, and the home page for taking all the inputs. I used a layout.html file to not repeat the codes for common elements. I then wrote the CSS and javascript files to make the website more interactive. All these files are stored in the sub-directory called static. I used SQLite for storing the user's data. In the end, I connected all the pages and restricted the user's access to the home page.

Chapter 3

User Manual

There are three main stakeholders in the website, the students, the instructors and the admins. The role of each is as follows:

3.1 For administrators:

There would be a single administrator of the portal. He would be given access to all the functionalities of the website(i.e. He would be made the Django Administrator). He would be able to view all the current instructors, add new instructors and delete existing instructors. He would be able to modify the details of the instructors as well.

The Administrator would also be allowed to add new students, assign them to existing instructors, reassign existing students to new instructors, change student details as well as delete enrolled students.

The Admin would be able to view the reports of all the students. He could do so by clicking on individual students and then viewing their report which would be containing the detailed reports of the student i.e. individual report of each assessment.

3.2 For instructors:

There would be one instructor for a group of students. The instructor would be tasked with providing the students the questions, the essay topics and gathering the responses on WhatsApp. Post which the instructor would have to export the WhatsApp chat in a file in .txt format. The instructor would be able to signup and login to the portal using his/her credentials, he would then be able to view his profile wherein he would be able to view his details and upcoming assignments/assessments.
The instructor could then navigate to either of two pages: the upload page; wherein he would

be able to upload the exported WhatsApp file along with the start date and end date of the evaluation as well as the name and the email ID of the student and then generate a report in either English language or Hindi language. The other page would be to view the performances of the students assigned to him. He would be able to view the individual performance of each student, how they performed over all the assessments as well the individual reports showing their report/errors in detail.

3.3 For students:

The students would form the basis of the entire web application, they would be asked to fill in textual answers of various questions/essays posed to them via any networking/messaging platform such as WhatsApp. They would then be asked to answer the questions/essays by the instructors and receive the answers using the same platform. This would be considered as the input to the Natural Language Processing Model.
The students would be able to signup and login to the portal using their credentials, on logging in they would be able to see their profile, with the name of the instructor showing along with their performances, i.e. how many essays have they answers as well as how they have performed overall, giving out a summary of all their evaluations. In addition, they would also be able to view their individual reports for each evaluation which would give an in-depth analysis of their performance in each evaluation using performance graphs, pie charts as well as pointing out individual errors in each sentence for the entire essay. This report would be generated in a pdf format.

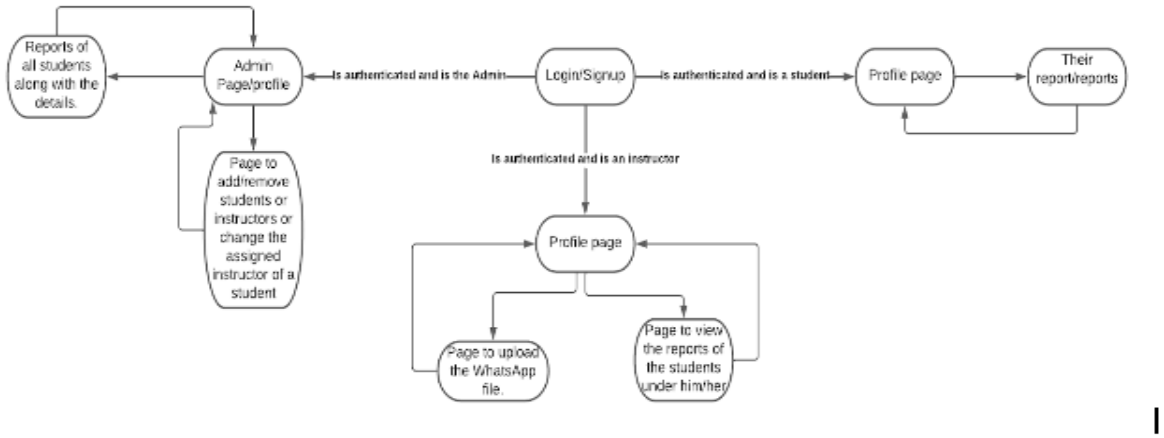

Figure 3.1: Layout of the website

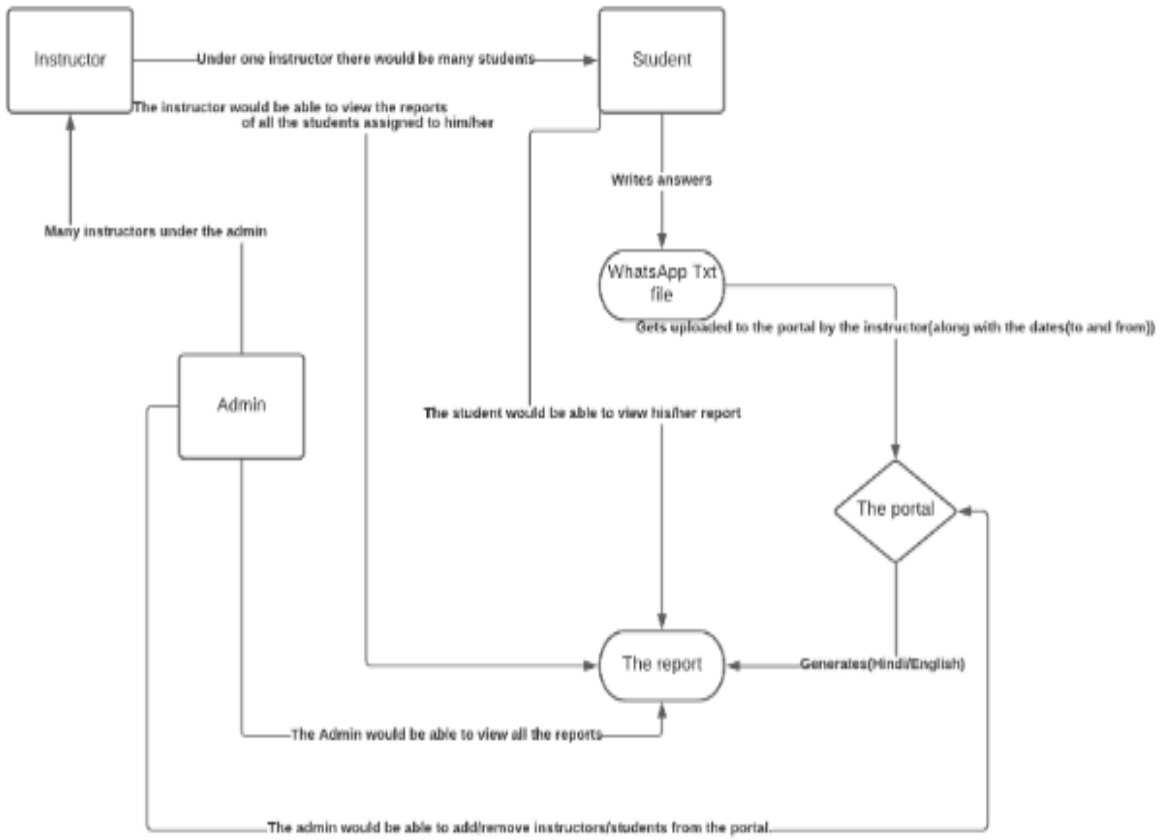

Figure 3.2: Diagram of all the stakeholder and their interactions

Chapter 4

Layout of the web application

The first screen of the website would contain the Signup page wherein the user could fill in their details as well as choose if they would like to sign up as a Student, an instructor or as the admin. Here, they would be required to fill in their First name, their last name as well as their email ID and set a password.

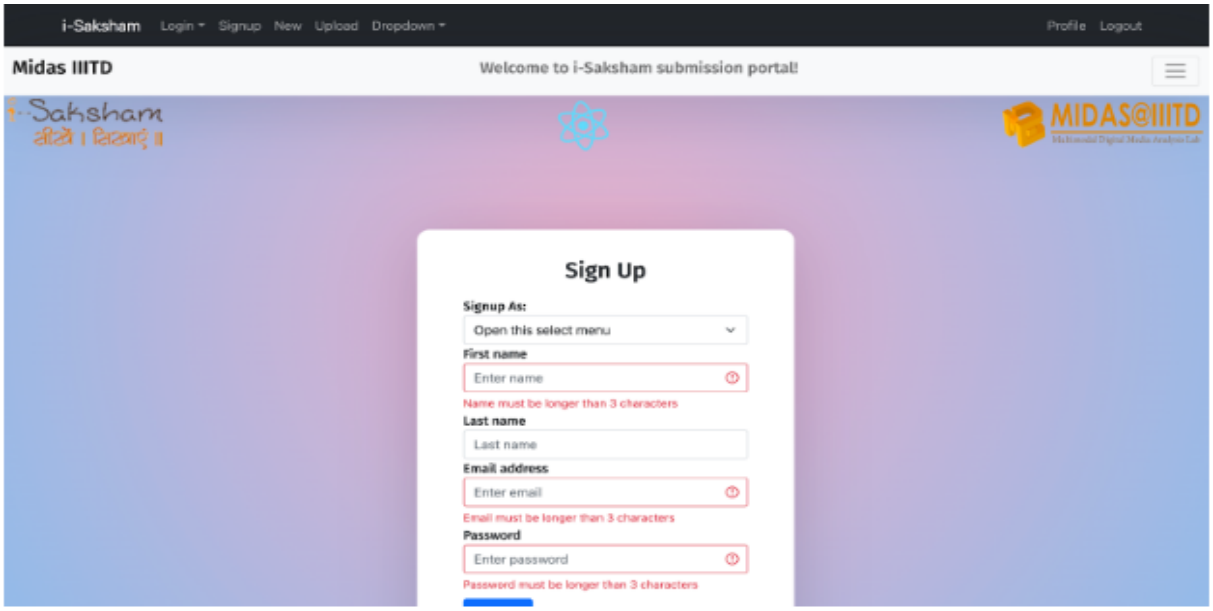

Figure 4.1: Signup page

Post Signup, the user would be required to login every time he/she wishes to visit the website. To login, the user would first have to choose their designation from a dropdown menu i.e. how would they like to login (as a student, an instructor or the admin) based on their choice they would be redirected to the appropriate login page.

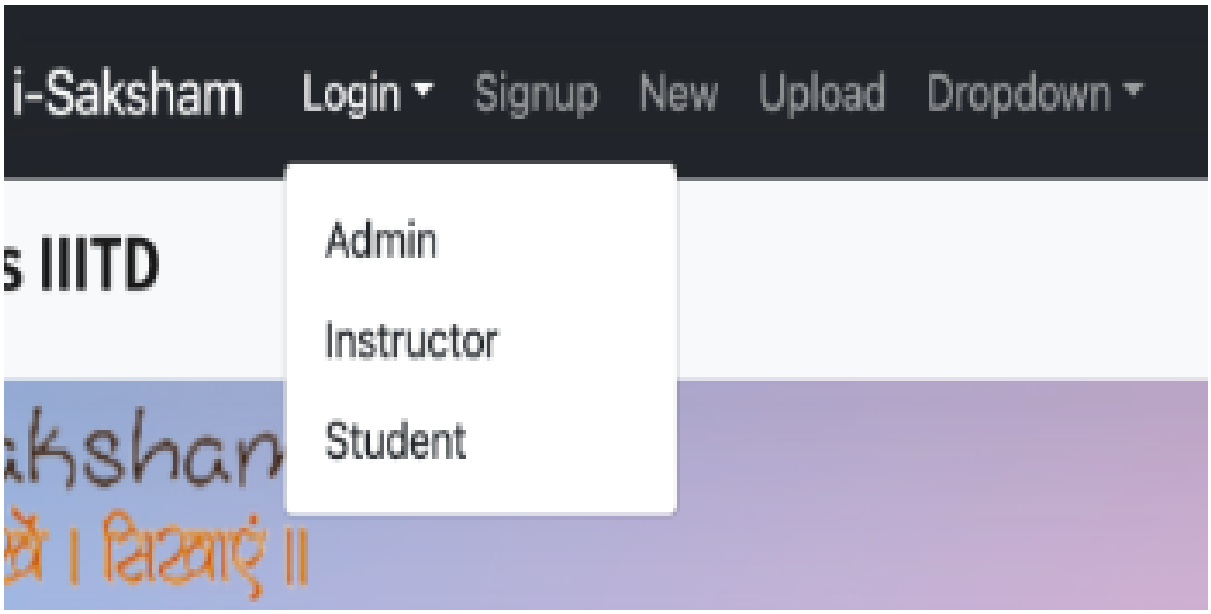

Figure 4.2: Designation wise login

On the login page, they would be asked to fill in the standard, their username and password.

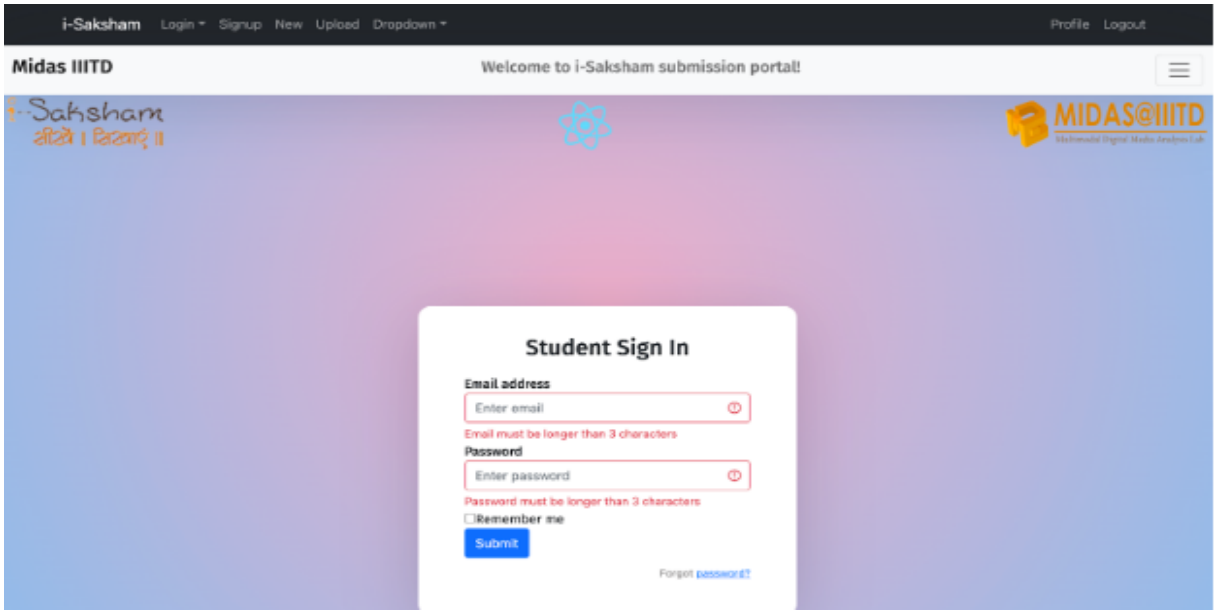

Figure 4.3: Student sign in page

Once the user has signed up on the portal he/she would redirected to the profile page(this would be common to all the designations i.e. the student, the instructor or the admin).

As a student, he/she would be able to view their details as well as view the detailed reports of all the assessments/assignments they have taken up so far.

As an instructor, he/she would be able to view their dashboard(their details) as well as navigate to the upload page(the page to upload the exported WhatsApp files) or the page wherein he/she could view the students under him/her and their detailed reports(of each assessment).

As the administrator, he/she would be able to view their details as well as an option to navigate to the page wherein he/she could view the reports of all the students or the page to make modifications to the students or instructors enrolled on the platform.

For demonstration purpose here is the instructor profile page:

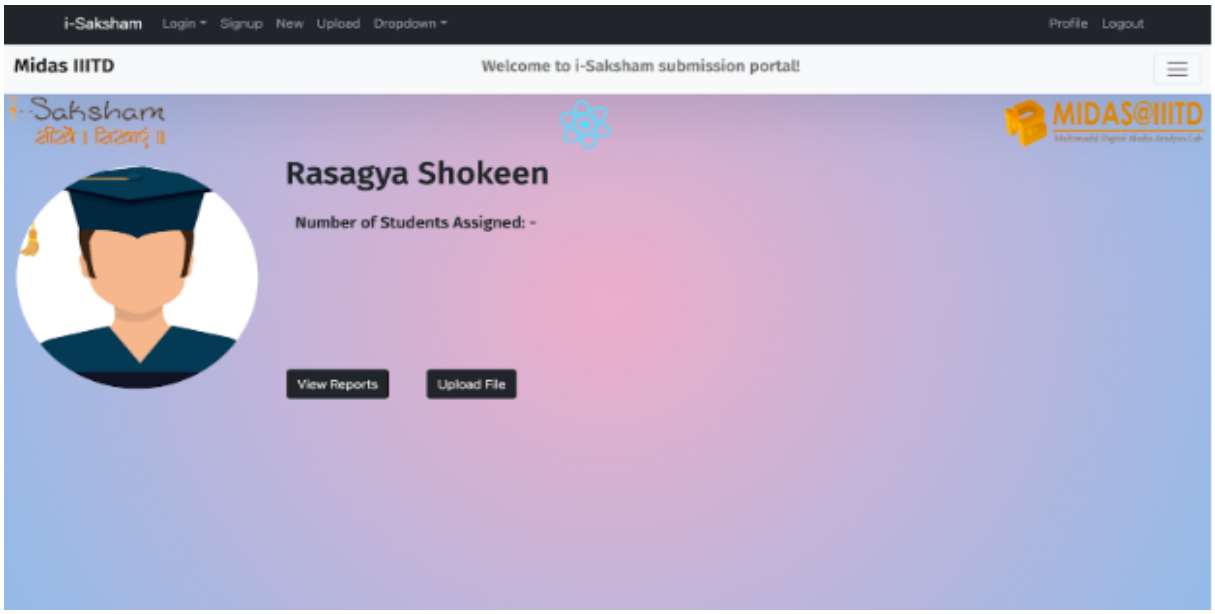

Figure 4.4: Profile page

The Hamburger Menu containing more details of the user:

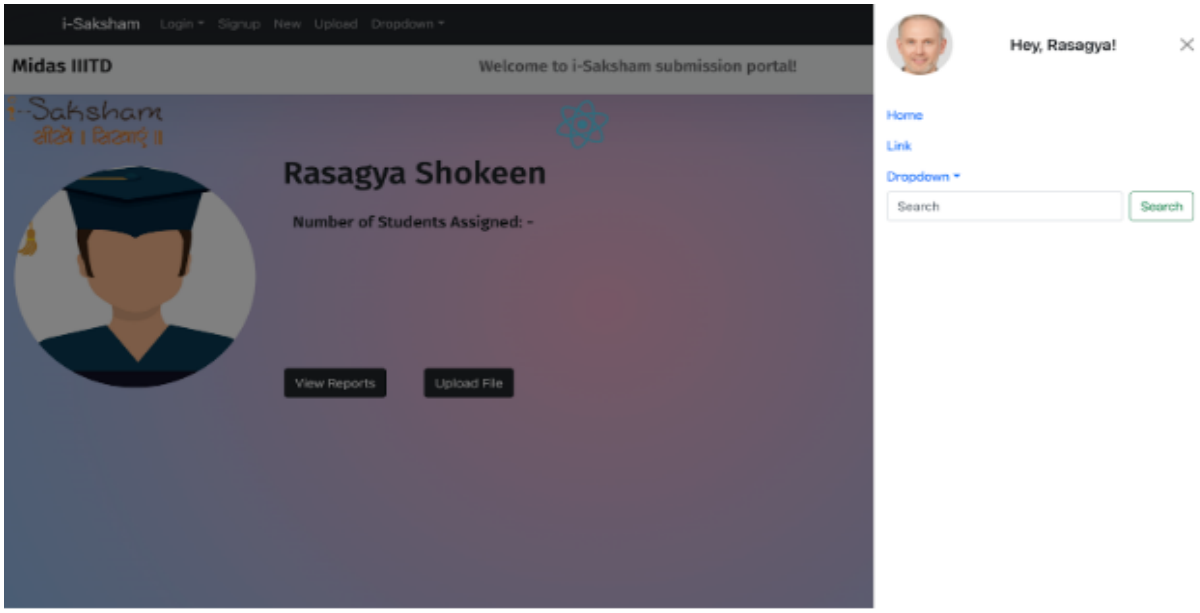

Figure 4.5: Hamburger menu

The instructor can then view the reports of all the students assigned to him, by clicking on the view reports button. Here the instructor would be able to click on individual student to view their detailed reports.

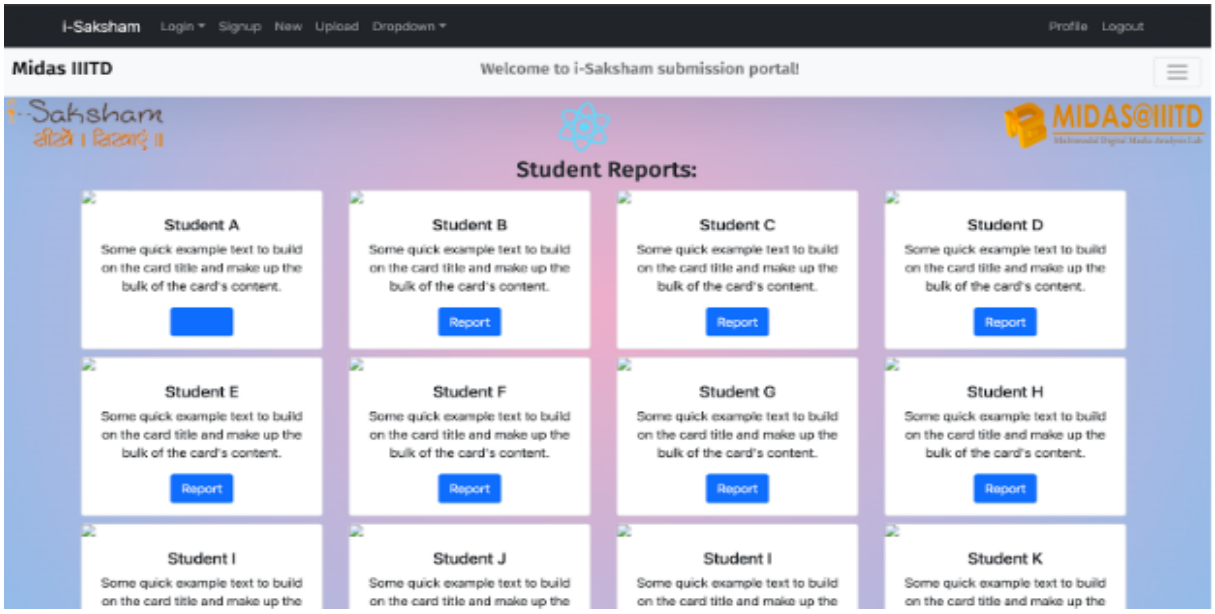

Figure 4.6: Student reports page

The instructor could also choose the other option of navigating to the upload page to upload the exported WhatsApp txt files of the student along with other details and choose whether he/she wants to generate the report in Hindi or English language.

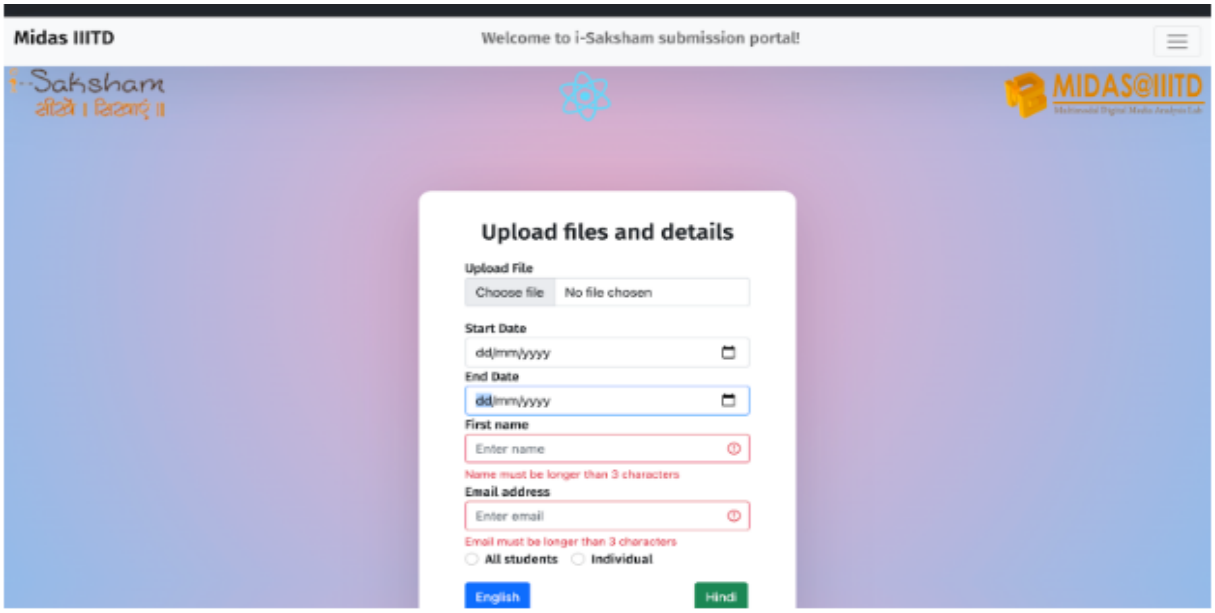

Figure 4.7: .txt file upload page

Here is a sample of the hindi report generated by the portal using NLP models and techniques.

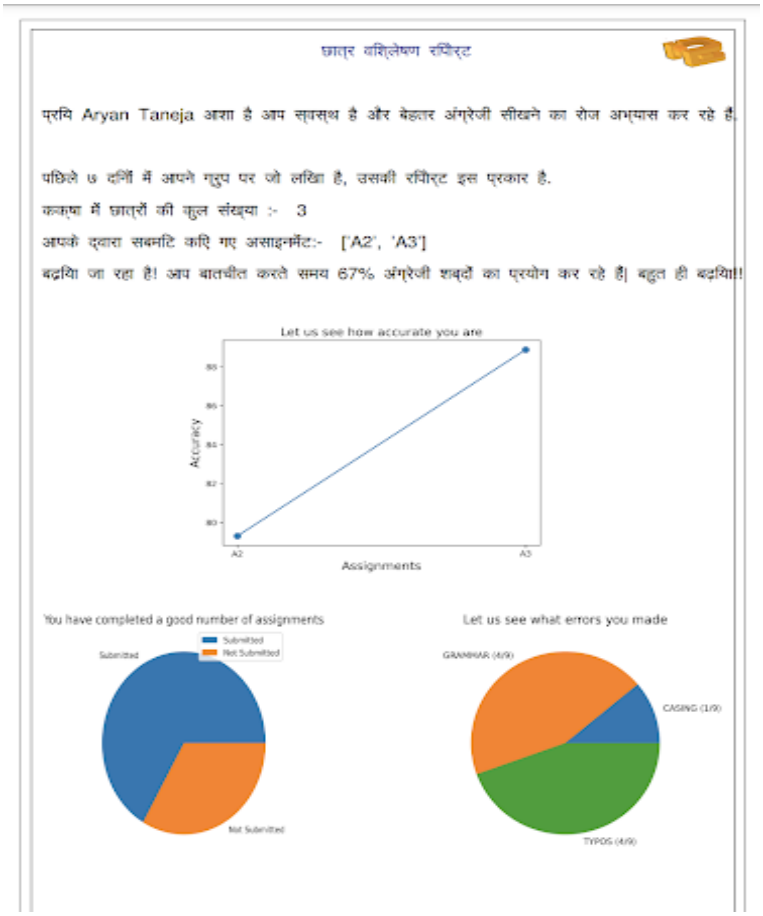

Figure 4.8: NLP generated report

Chapter 5

Conclusion and Future Work

Although, feedback of the new application was not possible at this point in time. Certain pointers were given out by the Head of i-Saksham, Mr. Aditya Tyagi sir, he wanted the application to have an option to generate the report in Hindi language as per the will and comfort of the administrator/student. He also requested for the creation of an interactive and frank report with various graphs, charts, greetings, salutations etc. Additional features which may be added to the application include a way for the students to view their report without the administrator's credentials and it can be explored in the future expansion of the website.
(As on the end of the first semester)
The backend framework of the website needs to be updated with the needs of the portal. Presently the frontend of the website has been completed with some parts of the Django backend being implemented as well. The NLP model could be further developed to include other languages such as Spanish, Mandarin etc. as well. This would help broaden the user base of the website increasing from the underprivileged to students all over the globe. More functionalities could be added to website as well with the number of stakeholders increasing.