A Major Project Report
on
Descriptive Online Examination Automation Using Natural Language Processing Techniques

Submitted in partial fulfilment of the
Requirements for the award of degree of

Bachelor of Technology
in
Computer Science and Engineering

By

M Suhas Rao
18H61A05M2

Under the Guidance of
Mr. Madar Bandu
Assistant Professor

Department of Computer Science and Engineering
ANURAG GROUP OF INSTITUTIONS
(Formerly CVSR College of Engineering)
(An Autonomous Institution, Approved by AICTE and NBA Accredited)
Venkatapur (V), Ghatkesar (M), Medchal(D)., T.S-500088
(2018-2022)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project entitled **"Descriptive Online Examination Automation Using Natural Language Processing Techniques"** being submitted by **M Suhas Rao** bearing the Hall Ticket number **18H61A05M2** in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag Group of Institutions** (**Formerly CVSR College of Engineering**) is a record of bonafide work carried out by him under my guidance and supervision from January 2022 to May 2022.

       The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

Internal Guide   **External examiner**
Mr. Madar Bandu,
Assistant Professor

                     Dr.G.Vishnu Murthy,
            Professor & Head, Dept. of CSE

ACKNOWLEDGEMENT

It is my privilege and pleasure to express profound sense of respect, gratitude and indebtedness to my guide **Mr. Madar Bandu, Assistant Professor**, Dept. of Computer Science and Engineering, Anurag Group of Institutions (Formerly CVSR College of Engineering), for his indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

I express my sincere gratitude to **Dr. G. Vishnu Murthy, Professor & Head**, Department of Computer Science and Engineering, Anurag Group of Institutions (Formerly CVSR College of Engineering), for his suggestions, motivations and co-operation for the successful completion of the work.

I extend my sincere thanks to **Dr.V. Vijaya Kumar**, **Dean,** Research and Development, Anurag Group of Institutions, for his encouragement and constant help.

M Suhas Rao
18H61A05M2

DECLARATION

I hereby declare that the project work entitled "**Descriptive Online Examination Automation Using Natural Language Processing Techniques**" submitted to the **Anurag Group of Institutions(Formerly CVSR College of Engineering)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B.Tech)** in Computer Science and Engineering is a record of an original work done by me under the guidance of **Mr. Madar Bandu, Assistant Professor** and this project work have not been submitted to any other university for the award of any other degree or diploma.

M Suhas Rao
18H61A05M2

Date:

ABSTRACT

This world has seen a lot many examination portals that are deployed over several servers which are used to conduct online examination for various purposes among which some may include conducting a test for entrance examinations, or Olympiads at national and international level and while some portals are designed to conduct a test for placement purposes. But what we have seen is that mostly all the portals are designed to conduct tests that contain multiple choice questions. Here our aim is not to work on the technology that is already existing, rather some technology that is very rare. Here we talk of the descriptive online examination system. Multiple choice questions are easy to deal as they have a question, a few options and a field in the same question that stores the correct option in the database. While in the case of descriptive questions it is not so. It brings in or uses the concepts of Natural Language Processing or NLP to assign marks to answers. Answers are nothing but strings and the job of the model is to do some operations on the answer string such that it can assign the correct marks to answers written by the examinee. The data is basically collected from a descriptive online examination system. Further, it is analyzed and the designed model assigns accurate marks to the answers for the question.

CONTENTS

1. Introduction

1.1 Motivation

While the offline examination system has a big disadvantage but are not getting replaced at a bigger scale because new online examination system features only multiple-choice type of question's while most of the exams contain descriptive question for which multiple choice answers do not work and hence, they are not that compatible and efficient to replace it at a larger level. We all know that if we have to remove a universally accepted system, the new system should not be just good, rather it should be able to make a quality difference so that the organizations accept it.

1.2 Problem Definition

In the proposed model we are taking the online examination system to a new level by enabling the examinee to write descriptive answers which will get evaluated on their own i.e., automating the entire offline examination system with the efficiency of computing having no human error involved, this can be done using NLP or Natural Language Processing. The evaluated answers will be stored in the database and they can be viewed anytime and a particular student profile will be maintained for better evaluation of the student.

1.3 Objective of the Project

This will be a huge boost to the online examination system as this will allow it to overcome its biggest con and it will also help the online examination system to stretch its paw even in the half-yearly or annual examination conducted by schools or college for evaluating the profile of the student. This will have instant benefits like the system will relieve the burden of the teachers and professors of checking copies and in return they can be more productive with their time in teaching things, this will also eliminate biasing in answer script checking and will have leased space for any human error.

2. Literature Survey

1. An automatic classifier for exam questions in Engineering: A process for Bloom's taxonomy

AUTHORS: K. Jayakodi, M. Bhandara and I. Perera

Assessment is an essential activity to achieve the objective of the course being taught and to improve the teaching and learning process. There are several educational taxonomies that can be used to assess the efficacy of assessment in engineering learning by aligning the assessment tasks in line with the intended learning outcomes and teaching and learning activities. This research is focused on using a learning taxonomy that fits well for computer science and engineering to categorize and assign weights to exam questions according to the taxonomy levels. Existing Natural Language Processing (NLP) techniques, Wordnet similarity algorithms with NLTK and Wordnet package were used and a new set of rules were developed to identify the category and the weight for each exam question according to Bloom's taxonomy. Using the result, the evaluators can analyze and design the question papers to measure the student knowledge from various aspects and levels. Prior evaluation was conducted to identify most suitable NLP preprocessing techniques to the context. A sample set of end semester examination questions of the Department of Computer science and Engineering (CSE), University of Moratuwa was used to evaluate the accuracy of the question classification; weight assignment and the main category assignment were validated against the manual classification by a domain expert.

2. Detection of users suspected of using multiple user accounts and manipulating evaluations in a community site

AUTHORS: N. Ishikawa, K. Umemoto, Y. Watanabe

Some users in a community site abuse the anonymity and attempt to manipulate communications in a community site. These users and their submissions discourage other users, keep them from retrieving good communication records, and decrease the credibility of the communication site. To solve this problem, we conducted an experimental study to detect users suspected of using multiple user accounts and manipulating evaluations in a community site. In this study, we used messages in the data of Yahoo! chiebukuro for data training and examination.

3. Automated Online Exam Proctoring", IEEE Transactions on Multimedia.

AUTHORS: Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu

This study found journalists use government sites most often to retrieve information. Problems include difficulty with verification, unreliable information and lack of contact information.

4. Design of Paperless Examination System for Principles of Database Systems

AUTHORS: G. Zhang, and H. Ke

Paperless examination is an important role of modern education, which can effectively reduce the teachers' workload and improve work efficiency. However, the current paperless examination system mainly deals with the objective questions, it is almost impossible to deal with subjective questions such as programming languages, particular in SQL. There is no such practical system as far as know. This article describes a novel SQL-based paperless examination system, including objective questions as well as SQL programming questions

5. Task Based Automatic Examination System for Sequenced Test

AUTHORS:S. Luo, J. Hu and Z. Chen

Computer greatly influences our educational environment. Over the last years, automatic computer examination systems have been widely used for computer-based tests. But these systems are based on traditional question-answer examination style which is not fit for the sequenced test. The sequenced test should consider the context of the examinee, e.g., the order of questions or the permissions of the examinee, to grade an examinee. In this paper, we propose an effective and practical automatic examination architecture based on task. The task is abstracted from the examination process and can meet the requests of the sequenced test, such as order and dependency. At the end of the paper, we implement an automatic examination system based on task for the stake test which proves quite efficient in practice.

3. Analysis

3.1 Existing System

This world has seen a lot many examination portals that are deployed over several servers which are used to conduct online examination for various purposes among which some may include conducting a test for entrance examinations, or Olympiads at national and international level and while some portals are designed to conduct a test for placement purposes. But what we have seen is that mostly all the portals are designed to conduct tests that contain multiple choice questions.

3.1.1 Disadvantages

❖ offline examination system

❖ online examination system features only multiple-choice type of question's

❖ Low Efficiency

3.2 Proposed System

In the proposed model we are taking the online examination system to a new level by enabling the examinee to write descriptive answers which will get evaluated on their own the evaluated answers will be stored in the database and they can be viewed anytime and a particular student profile will be maintained for better evaluation of the student. Talking about the technology used in order to build such a model for evaluating descriptive answers, NLP or Natural Language Processing is has a great role to play.

3.2.1 Advantages

❖ High Efficiency.

❖ online descriptive examinations system.

❖ Answers are evaluated at that moment itself and the student can see the solutions and can correct the mistakes or errors committed while appearing for the exam.

3.3   Software Requirement Specification

3.3.1   Purpose

The purpose of this document is to build a Descriptive Online Examination Automation System Using Natural Language Processing to give automate the process of examination by analysing the questions and answers and calculating the marks.

3.3.2   Scope

The purpose of the Descriptive Online Examination Automation System Using Natural Language Processing is automating the process of examination by analysing the questions and answers and calculating the marks. Data pre-processing, the initial part of the project is to understand implementation and usage of various python inbuilt modules. The above process helps us to understand why different modules are helpful rather than implementing those functions from scratch by the developer. These various modules provide better code representation and user understandability. The following libraries are used such as NumPy, SciPy pandas, csv, sklearn, matplotlib, sys, re, emoji, nltk.

Exploratory data analysis, first step in this to apply a Text analysis algorithm i.e., BERT and TF-IDF algorithms which for evaluating answers for questions that are descriptive in nature. Keywords are stored for every question and on the basis their occurrence in the answer string, the examinee is allotted marks. In order to perform this in Python language, there exists a library that makes it a little easier for the algorithm developers to perform string manipulations. The name of the library used is NLTK which is specifically designed for python to work on NLP. It is discussed in detail in the section where the algorithm is discussed.

3.3.3 Overall Description

Talking about the technology used in order to build such a model for evaluating descriptive answers, NLP or Natural Language Processing is has a great role to play. NLP can do a lot of innovative jobs like predicting if a message or an email is a spam or a ham, the quality search that we can do on shopping websites like amazon and flipkart in order to search for different categories of items that include kitchen utensils, electronics gadget, apparels, food items and much more such products that are available online. The basic idea was that did anyone ever think of knowing how these search bars or how these ham spam classifications work? The answer to this question is that rarest of the rare people have tried getting into this and tried to know what the mechanism or the back-end work in order to give such powerful search results and such predictive classification techniques. For those who are not aware of the mechanism behind this, it's all just about playing with strings of characters, numbers and special characters or what we call as string manipulations to arrive at such results.

3.3.4 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

➤ ECONOMICAL FEASIBILITY

➤ TECHNICAL FEASIBILITY

➤ SOCIAL FEASIBILITY

3.3.4.1 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3.4.2 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3.4.3 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3.4.4 Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python IDE 3.7 version**
- **PyCharm - The Python IDE for Professional Developers**
- **Jupyter notebook**
- **Front-End – Html and CSS**
- **Database - SQLite**

3.3.4.5 Software Requirements

Minimum hardware requirements are very dependent on the particular software being developed by a given a thought Python / PyCharm / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system**   **: windows**
- **Processor**   **: minimum intel i3**
- **Ram**   **: minimum 4 gb**
- **Hard disk**   **: minimum 250gb**

3.3.4.5 Functional Requirements

1.Data Collection

2.Data Pre-processing

3.Training and Testing

4.Modelling

5.Predicting

3.3.4.6 Non-Functional Requirements

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

4. Design

4.1 UML diagrams

UML (Unified Modelling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. It was initially started to capture the behaviour of complex software and non-software system and now it has become an OMG standard. This tutorial gives a complete understanding on UML.

4.2 Methodology

The method of Descriptive Online Examination Automation is divided into different stages, the process includes gathering relevant data, prepare data, model building and Generate results and upload it to the website to obtain the results.
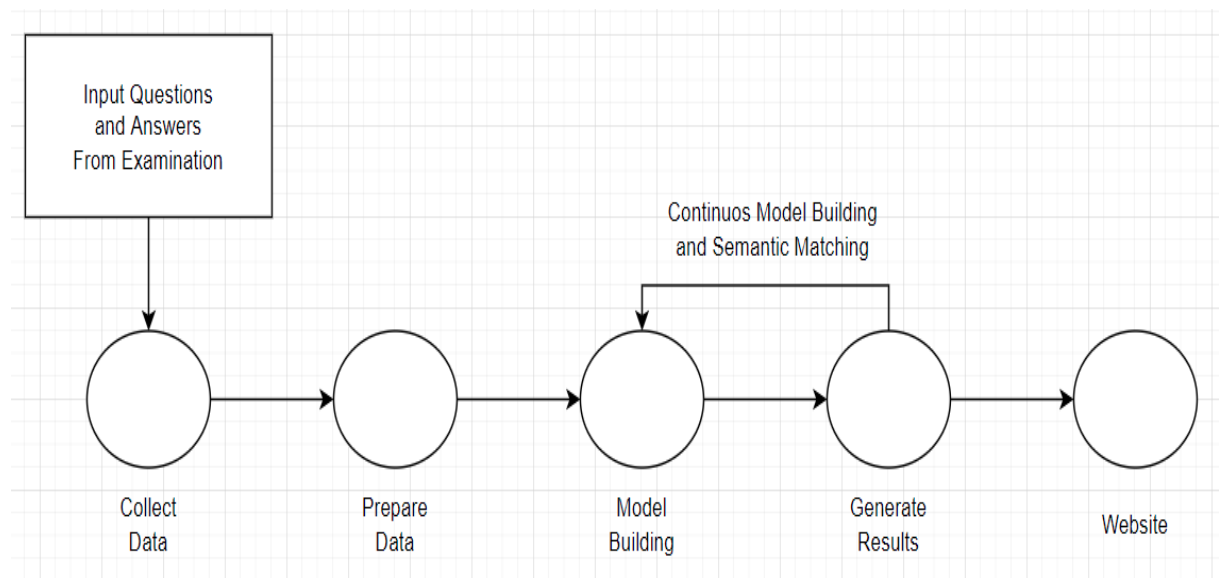
Fig 4.2

4.3 Project Architecture

Collecting data is the first step of the process flow which includes defining project, setting up the machine environment suitable for the development requirements and later understanding the data using different python libraries and machine learning techniques. Data Cleaning need to done on the data collected so that the analysis be very accurate for perfect results.

The questions and answers are upload by the examination which will be collected and the data extraction and analysis is done on the data using the tokenization algorithm

After the data extraction and cleaning is done we will be doing Data preparation using Stemming and Lemmitization algorithm where the tokens generated are separated and given separate branches based on the category of the words

Then the model training is done by using algorithms TF-IDF to generate the specific keywords and after that Using BERT algorithm the semantic matching is done.
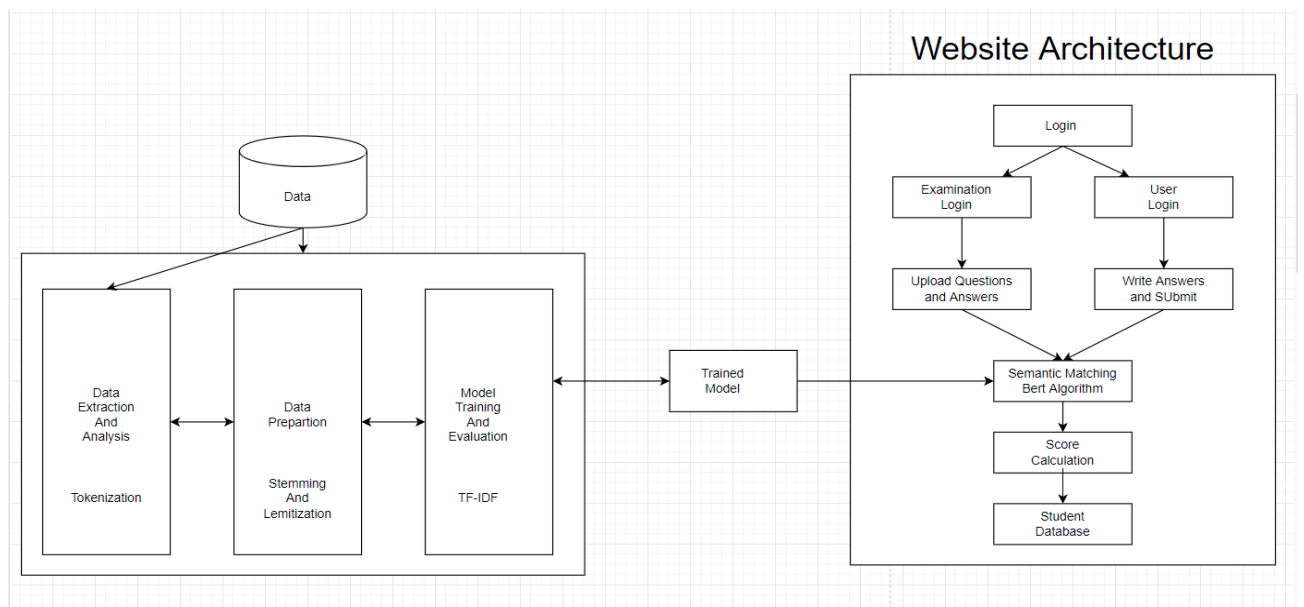

Fig 4.3

4.4 DATA FLOW DIAGRAM

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.
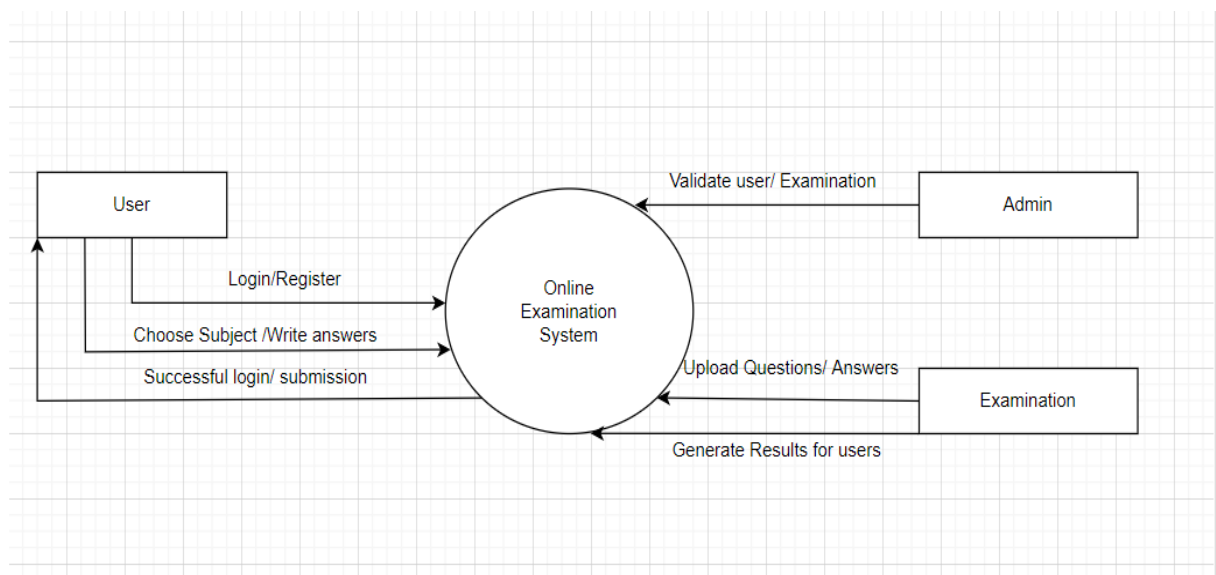

Fig 4.4

4.5 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.
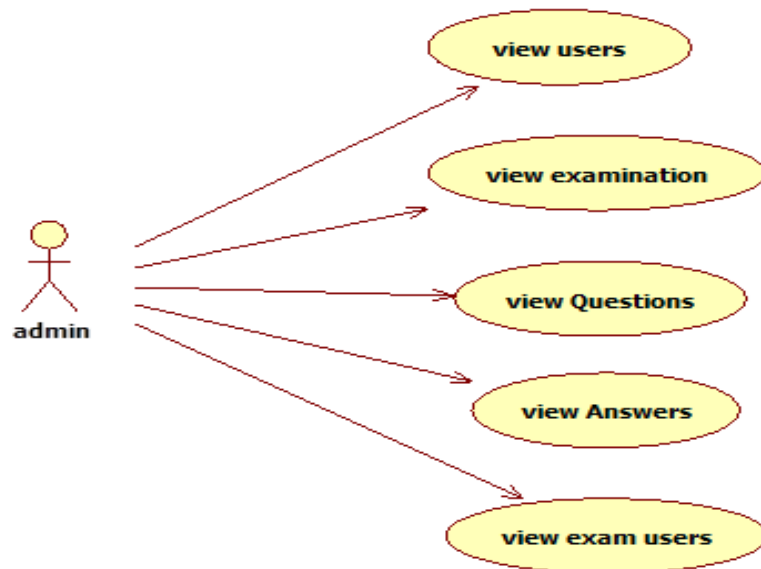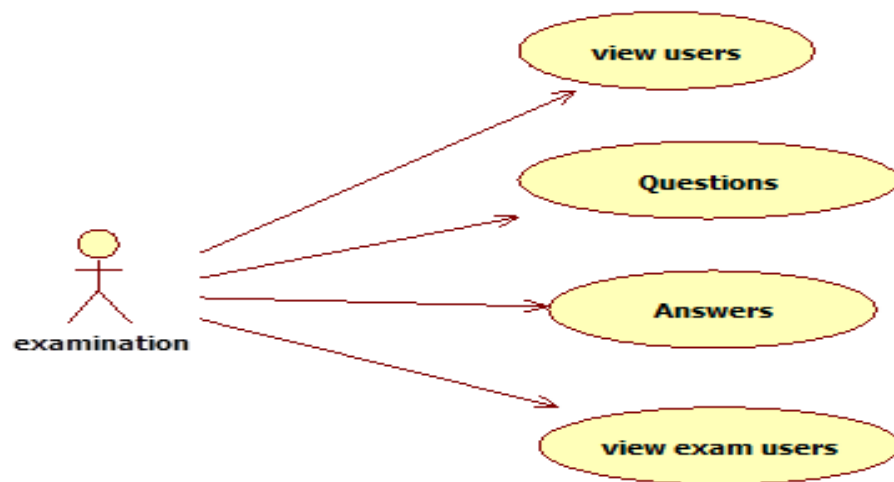
ADMIN:

Fig 4.5(A)

EXAMINATION:
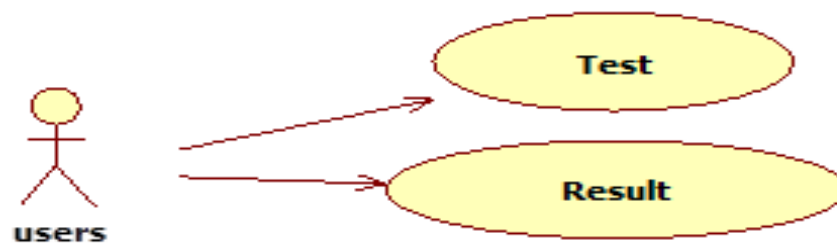

Fig 4.5(B)

USER:

FIG 4.5(C)

4.6 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, even scenarios, and timing diagrams.
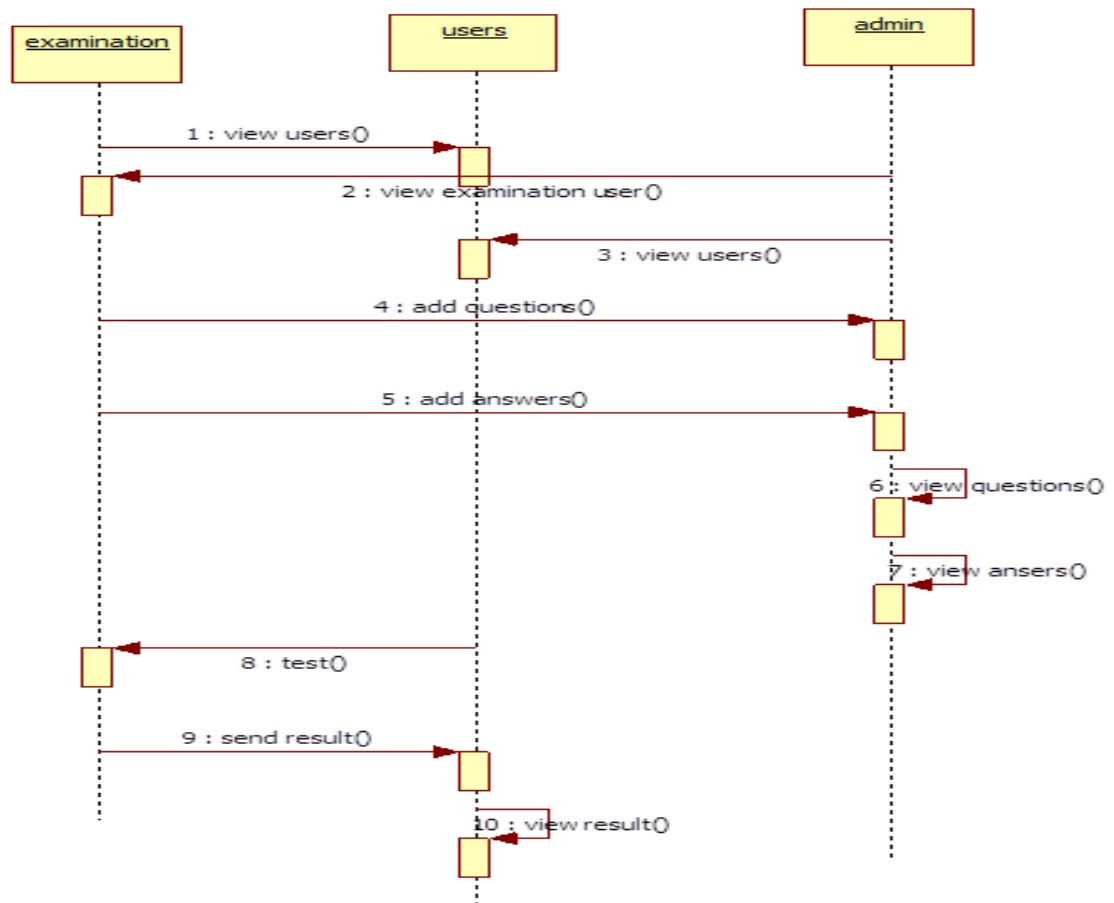

Fig 4.6

4.7 Activity Diagram

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.
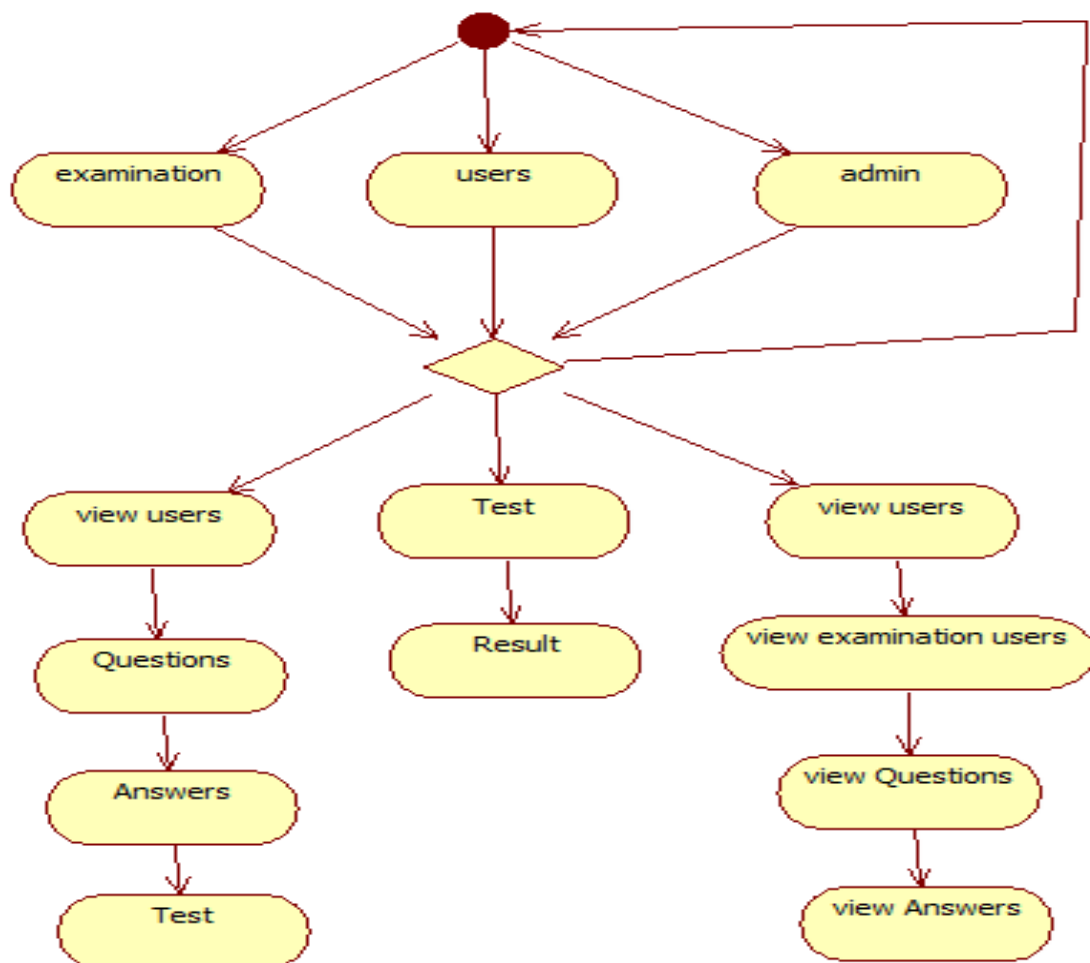

Fig 4.7

25

4.8 Component Diagram

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.
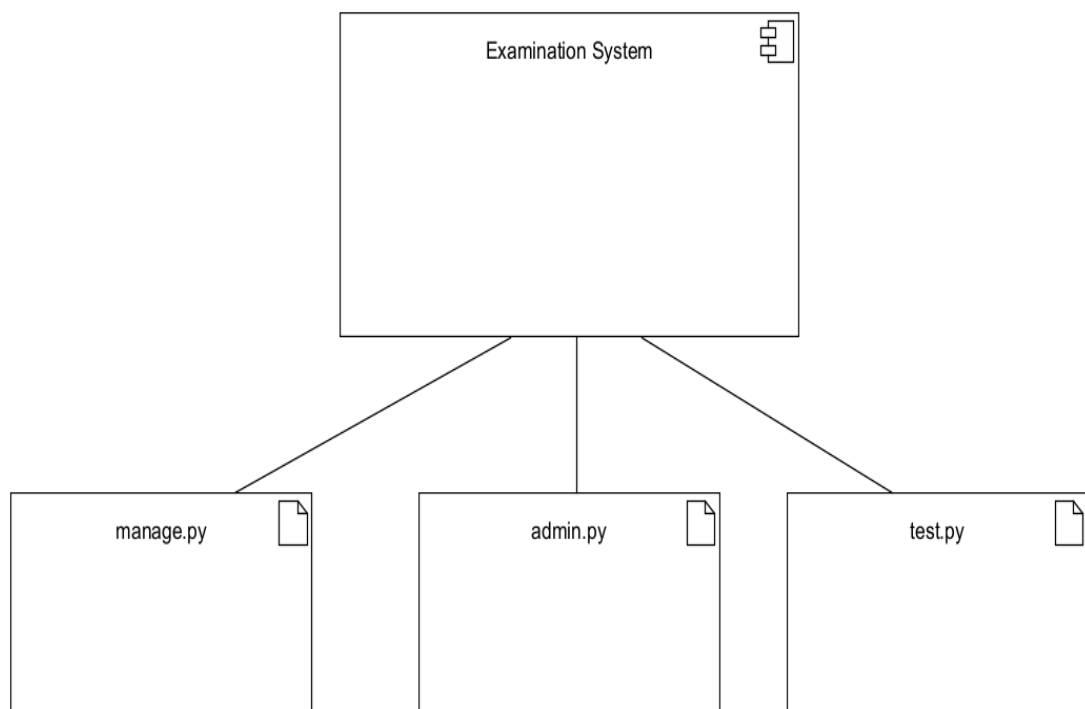

Fig 4.8

4.9 Deployment Diagram

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.
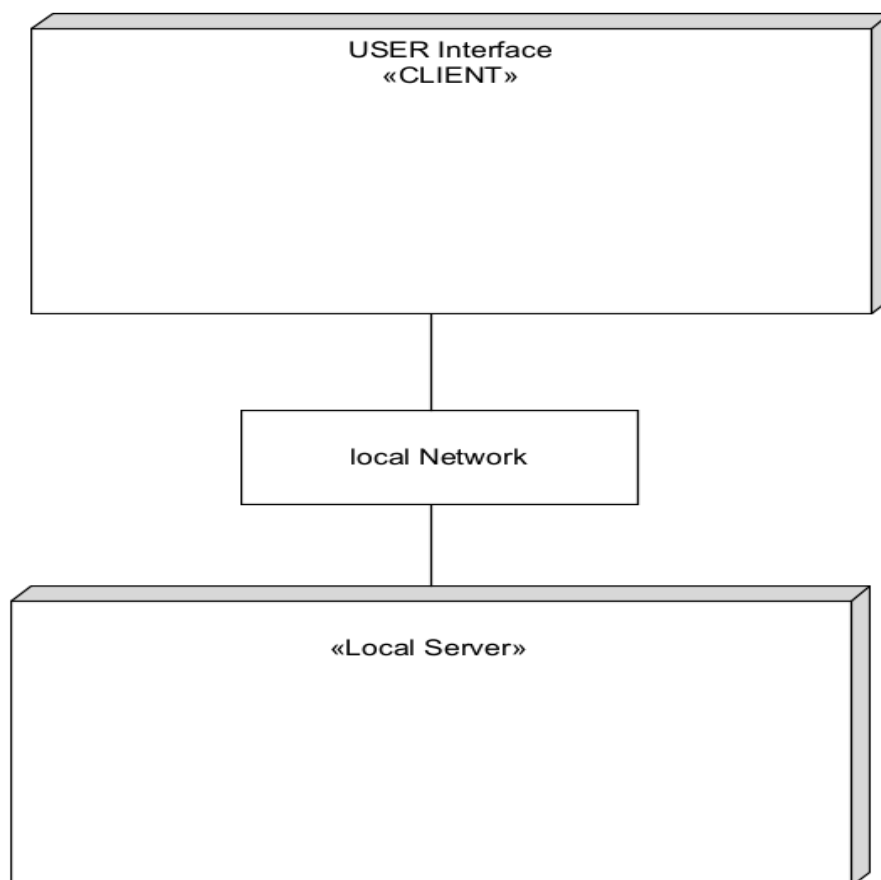

Fig 4.9

5. Implementation

5.1 Modules

MODULES:

- Examination

- User

- Admin

- Python and Data-analysis.

5.2 Module Description

5.2.1 Examination.py

Examcell conducts the exams on particular technologies. First, it will store all questions about particular technology into the database.

Then store all answers of every technology into the database. After, conducting the exam it will check the student answers with database answers. Finally by using nltk and nlp concept give the result to students.

5.2.2 User.py

The content of it may be overlapping with that of others, but it also has its own discipline characteristics and is constantly developing new theories and methods.user after login into the session they will choose their particular exam .

Then student will write the exam and submit the answers. After clicking test result automatically student will get the result based on their performance. all these concepts i.e., getting of student score by using nltk and nlp concepts.

5.2.3 App.py

Admin will give authority to Exam-cell and students.

In order to facilitate activate the Exam Cell and activate the students. the admin can see the details of Exam cell and students.

Admin can set the score of all students..

Python and data-analysis:

Python is an increasingly popular tool for data analysis. In recent years, a number of libraries have reached maturity, allowing R and Stata users to take advantage of the beauty, flexibility, and performance of Python without sacrificing the functionality these older programs have accumulated over the years.python focus on simplicity and readability, python it boasts a gradual and relatively low learning curve. This ease of learning makes an ideal tool for beginning programmers. Python offers programmers the advantage of using fewer lines of code to accomplish tasks than one needs when using older languages

5.3 Python Libraries

5.3.1 NumPy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python

5.3.2 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

5.3.3 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

5.3.4 RegEx

A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern RegEx can be used to check if a string contains the specified search pattern. Python has a built-in package called re, which can be used to work with Regular Expressions.

5.3.5 Natural Language Toolkit

NLTK stands for Natural Language Toolkit and it is suite of libraries and programs in Python for Natural Language Processing Tasks. It is one of the most widely used NLP Python libraries.

It can perform various NLP tasks like tokenization, stemming, POS tagging, lemmatization and classification to name a few.

5.3.6 Algorithms Used

TF-IDF stands for *term frequency-inverse document frequency* and it is a measure, used in the fields of Information Retrieval and machine learning, that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc) in a document amongst a collection of documents (also known as a corpus).

Overview of TF-IDF

TF-IDF can be broken down into two parts *TF (term frequency)* and *IDF (inverse document frequency)*.

What is TF (term frequency)?

Term frequency works by looking at the frequency of a *particular term* you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency:

- Number of times the word appears in a document (raw count).
- Term frequency adjusted for the length of the document (raw count of occurences divided by number of words in the document).
- Logarithmically scaled frequency (e.g. log(1 + raw count)).

BERT (Bidirectional Encoder Representations from Transformers) is a recent paper published by researchers at Google AI Language. It has caused a stir in the Machine Learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others.

BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. In the paper, the researchers detail a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible.

5.4 Introduction to Technologies Used

5.4.1 Python

Python is currently the most widely used multi-purpose, high-level programming language.Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

5.4.2 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License,and there is also Professional Edition with extra features – released under a proprietary license.

5.4.3 Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

5.4.4 HTML,CSS,JAVASCRIPT

- **HTML** provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and JavaScript.

- **CSS** is used to control *presentation, formatting, and layout*.

- **JavaScript** is used to control the *behavior* of different elements.

5.4.5 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

5.4.6 Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

5.4 Sample Code

5.4.1 Url.py:

```
from django.contrib import admin from django.urls import path from django.conf import settings

from django.conf.urls.static import static from user import views as user

from examination import views as examination

from ExaminationSystem import views as ExaminationSystem

urlpatterns = [

path('admin/', admin.site.urls), #path('hello/', teacher.hello), path('index/', user.index, name='index'),

path('userpage/', user.userpage, name='userpage'), path('userlogin/', user.userlogin, name='userlogin'), path('userregister/', user.userregister, name='userregister'),

path('userlogincheck/', user.userlogincheck, name='userlogincheck'), path('usertest/', user.usertest, name='usertest'),

path('examtest/', user.examtest, name='examtest'), path('usertestcheck/', user.usertestcheck, name='usertestcheck'), path('usersanswers/', user.usersanswers, name='usersanswers'),

path('examinationlogin/', examination.examinationlogin, name='examinationlogin'), path('examinationregister/', examination.examinationregister,

name='examinationregister'),

path('examinationcheck/', examination.examinationcheck, name='examinationcheck'), path('usersdata/', examination.usersdata, name='usersdata'),
```

```python
path('questioncell/', examination.questioncell, name='questioncell'), path('examquestions/',
examination.examquestions, name='examquestions'), path('qanswers/', examination.qanswers,
name='qanswers'), path('userresult/', examination.userresult, name='userresult'),
path('examresult/', examination.examresult, name='examresult'),

path('adminlogin/', ExaminationSystem.adminlogin, name='adminlogin'),

path('adminloginentered/', ExaminationSystem.adminloginentered,
name='adminloginentered'),

path('viewuserdata/', ExaminationSystem.viewuserdata, name='viewuserdata'),
path('examinationquestionsdata/', ExaminationSystem.examinationquestionsdata,

name='examinationquestionsdata'),

path('examinationanswersdata/', ExaminationSystem.examinationanswersdata,
name='examinationanswersdata'),

path('userstestdata/', ExaminationSystem.userstestdata, name='userstestdata'),
path('activateuser/', ExaminationSystem.activateuser, name='activateuser'),
path('examinationuserdata/', ExaminationSystem.examinationuserdata,

name='examinationuserdata'),

path('activateexamination/', ExaminationSystem.activateexamination,
name='activateexamination'),

path('logout/', ExaminationSystem.logout, name='logout'),

]
```

Mainsite:

User Views.py

```python
from django.contrib import messages from django.shortcuts import render from django.http import HttpResponse

from examination.models import questionsModel, ExaminationQuestionModel from user.forms import *

from user.models import user

def index(request):

return render(request,'index.html')

def userpage(request):

return render(request,'user/userpage.html')

def userlogin(request):

return render(request,"user/userlogin.html")

def userregister(request):

if request.method=='POST': form1=userForm(request.POST) if form1.is_valid():

form1.save()

return render(request, "user/userlogin.html")

#return HttpResponse("registreration succesfully completed")

return render(request, 'user/userquestions.html', {"object":check})

def examtest(request):
```

```python
if request.method=='POST':

email = request.POST.get('email') #question = request.POST.get('question') #answer = request.POST.get('answer') questions = request.POST.getlist('qtn[]')

answers = [request.POST['answer-{}'.format(q)] for q in questions] print("Kattama code ",questions,'==',answers)

questions = request.POST.getlist('answers') print("question",type(questions),questions)

answers = [request.POST['answer-{}'.format(q)] for q in questions] print("email",email,"question",questions,"answer",answers)

form1 = examtestForm(request.POST) if form1.is_valid():

form1.save()

return render(request, "user/userpage.html")

#return HttpResponse("registreration succesfully completed") else:

print("form not valied")

return HttpResponse("form not valied")

else:

form = examtestForm()

return render(request,"user/usertest.html",{"object":form})

def usersanswers(request):

if request.method=='POST': email = request.session['email']
```

subject = request.POST.get('subject') question1 = request.POST.get('question1') answer1 = request.POST.get('answer1') question2 = request.POST.get('question2') answer2 = request.POST.get('answer2') question3 = request.POST.get('question3') answer3 = request.POST.get('answer3') question4 = request.POST.get('question4') answer4 = request.POST.get('answer4') question5 = request.POST.get('question5') answer5 = request.POST.get('answer5') question6 = request.POST.get('question6') answer6 = request.POST.get('answer6') question7 = request.POST.get('question7')

examination View.py:

from django.contrib import messages from django.shortcuts import render import nltk

Create your views here.

from examination.forms import examinationForm, examinationanswerForm, examinationquestionForm

from django.http import HttpResponse

from examination.models import * from user.models import *

def examinationlogin(request):

return render(request,"examination/examinationlogin.html")

def examinationregister(request): if request.method=='POST':

form1=examinationForm(request.POST)

def examinationcheck(request): if request.method == "POST":

email = request.POST.get('uname') pswd = request.POST.get('upasswd')

print("Email = ", email, ' Password = ', pswd) try:

```python
check = examination.objects.get(email=email,passwd=pswd) status = check.status

print('Status is = ', status) if status == "Activated":

request.session['id'] = check.id #request.session['name'] = check.name request.session['email']
= check.email print("User id At", check.id, status)

return render(request, 'examination/examinationpage.html', {}) else:

messages.success(request, 'Your Account Not at activated') return render(request,
'examination/examinationlogin.html')

# return render(request, 'user/userpage.html',{}) except Exception as e:

print('Exception is ', str(e)) pass

messages.success(request, 'Invalid Email id and password') return render(request,
'examination/examinationlogin.html')

def usersdata(request): object = user.objects.all()

return render(request,"examination/userdata.html",{"object":object]

"""def examcell(request):

if request.method == 'POST':

p = ''.join(e for e in remove if e.isalnum()) print("remove punctivations",p)

object1 = UserTestModel.objects.get(email=email) print("object1",object1.answer1)

s1 = ''

s1 = s1+object1.answer1+" "+object1.answer2+" "+object1.answer3+" "+object1.answer4+"
"+object1.answer5+" "+object1.answer6+" "+object1.answer7+" "+object1.answer8+"
"+object1.answer9+" "+object1.answer10
```

```python
print("student answers",s1) words = s1.split()

nltk = ' '.join(sorted(set(words), key=words.index)) print("remove duplicate words", nltk)

p1 = ''.join(e for e in nltk if e.isalnum()) print("remove punctivations for users", len(p1)) m = len(p)+len(p1)

if p == p1:

c = (len(p1)/m)*100 print("percentage",c)

else:

c = ( len(p1) / m) * 100 print("percentage", c)

return render(request, "examination/examinationpage.html")
```

examination model.py:

```python
from django.db import models

# Create your models here.

class examination(models.Model):

firstname = models.CharField(max_length=50,default="", editable=True) lastname = models.CharField(max_length=50,default="", editable=True) email = models.EmailField()

passwd = models.CharField(max_length=40)

mobileno = models.CharField(max_length=50, default="", editable=True) qualification = models.CharField(max_length=40)

city = models.CharField(max_length=40)

status = models.CharField(max_length=40,default="", editable=True)
```

41

examination forms.py

```python
from django import forms

from django.core import validators

from examination.models import examination, ExaminationQuestionModel,

ExaminationAnswerModel

def name_check(value):

if value.isalpha()!=True:

raise forms.ValidationError("only string are allowed")

class examinationForm(forms.ModelForm):

firstname       =       forms.CharField(widget=forms.TextInput(),       required=True,
max_length=100,validators=[name_check])

lastname       =       forms.CharField(widget=forms.TextInput(),       required=True,
max_length=100,validators=[name_check])

passwd = forms.CharField(widget=forms.PasswordInput(), required=True, max_length=100)

email   =   forms.CharField(widget=forms.TextInput(),   required=True)   mobileno=
forms.CharField(widget=forms.TextInput(), required=True,

max_length=10,validators=[validators.MaxLengthValidator(10),validators.MinLengthVal
idator(10)])

qualification = forms.CharField(widget=forms.TextInput(), required=True, max_length=100)

city = forms.CharField(widget=forms.TextInput(), required=True, max_length=100) status =
forms.CharField(widget=forms.HiddenInput(), initial='waiting',
```

max_length=100)

ExaminationSystem views.py

```python
from django.shortcuts import render from 54jango.http import HttpResponse

from examination.models import examination, ExaminationQuestionModel,

ExaminationAnswerModel

from user.models import user, UserTestModel

def adminlogin(request):

return render(request, "admin/adminlogin.html")

def adminloginentered(request): if request.method == 'POST':

uname=request.POST['uname'] passwd=request.POST['upasswd']

if uname == 'admin' and passwd == 'admin@2020': return render(request,"admin/adminloginentered.html")

else:

return HttpResponse("invalied credentials")

def logout(request):

return render(request,'index.html')

def viewuserdata(request): object = user.objects.all()
```

```python
return render(request,"admin/viewuserdata.html",{"object":object})

def activateuser(request):

    if request.method =='GET': uname=request.GET.get('pid') print(uname) status='Activated'

    print("pid=",uname,"status=",status)        user.objects.filter(id=uname).update(status=status)
    object=user.objects.all()

    return render(request,"admin/viewuserdata.html",{"object":object})

def examinationuserdata(request): object = examination.objects.all()

    return render(request,"admin/examinationuserdata.html",{"object":object})

def activateexamination(request): if request.method =='GET':

    uname=request.GET.get('pid') print(uname) status='Activated'

    print("pid=",uname,"status=",status)
    examination.objects.filter(id=uname).update(status=status) object=examination.objects.all()

    return render(request,"admin/examinationuserdata.html",{"object":object})

def examinationquestionsdata(request):

    object = ExaminationQuestionModel.objects.all()

    return render(request,"admin/examinationquestionsdata.html",{"object":object})

def examinationanswersdata(request):

    object = ExaminationAnswerModel.objects.all()
```

6. Test Cases

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1.	Examination login	If username and password is correct then it will getting valid page.	Pass	If username or password is not correct (or) admin not activated.
2.	User Registration	If data user is registered successfully.	Pass	If data user is not registered.
4.	User login	Username and password is correct then getting valid page.	Pass	If username or password is wrong (or) admin not activated
5.	View users	User registration details available.	Pass	If user is not registered
6.	Upload questions	Examinar upload questions	Pass	Examinar not upload questions
7.	Upload answers	Examinar upload answers	Pass	Examinar not upload answers
8.	Test	User write the exam	Pass	User not attend the exams

| 9 | score | Examinater send the score to the test | pass | Examinar not assign the scor to the test |
10	View examinar	Examination registration details available.	pass	If examination is not registered
11	View question	If examination user upload questions	pass	Examination user not upload the questions
12	View Answers	If examination user upload answers	pass	Examination user not upload the answers

7. Screen Shots
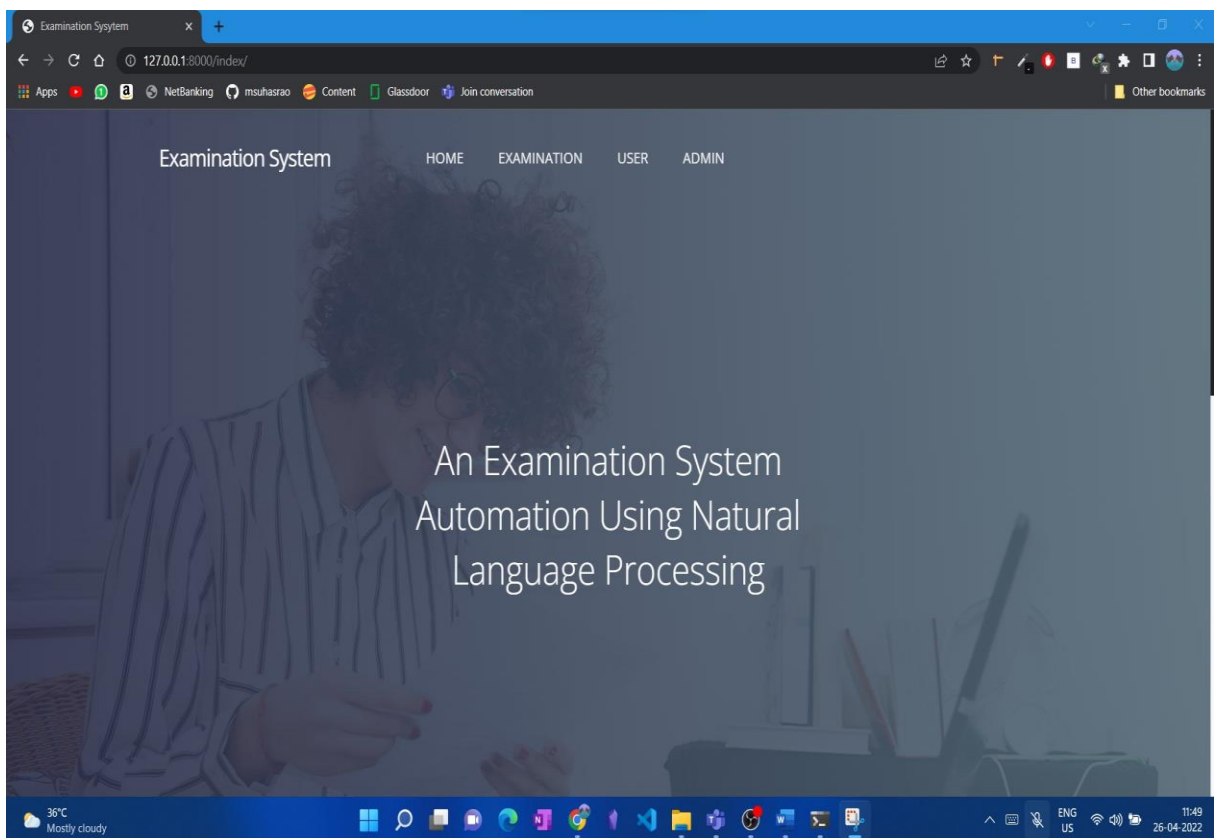
Fig 7.1 Home page
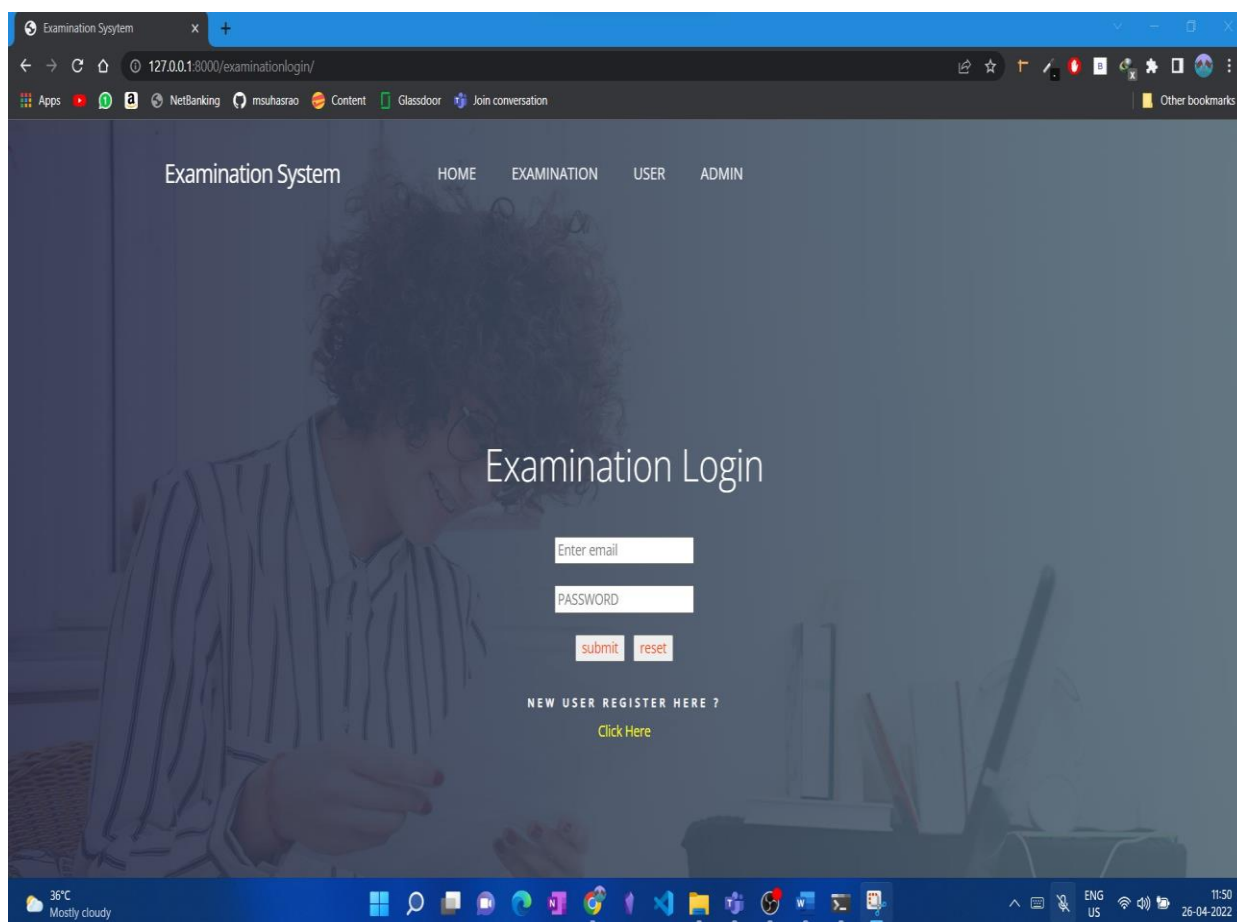
It consists of 3 tabs examination, user and admin

Fig 7.2 Examination login

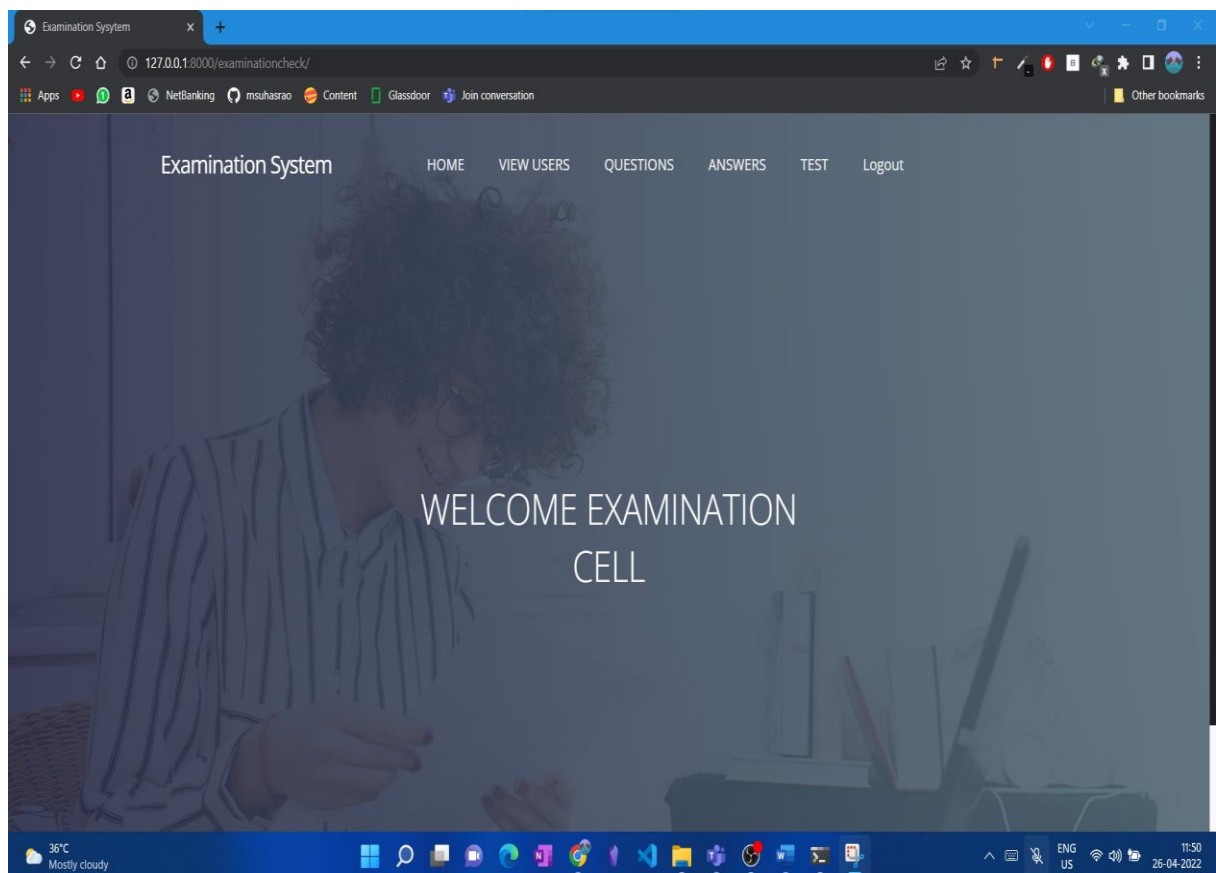**It consists of Examination login where the examiner logins to their account**

Fig 7.3 Home page of Examination login

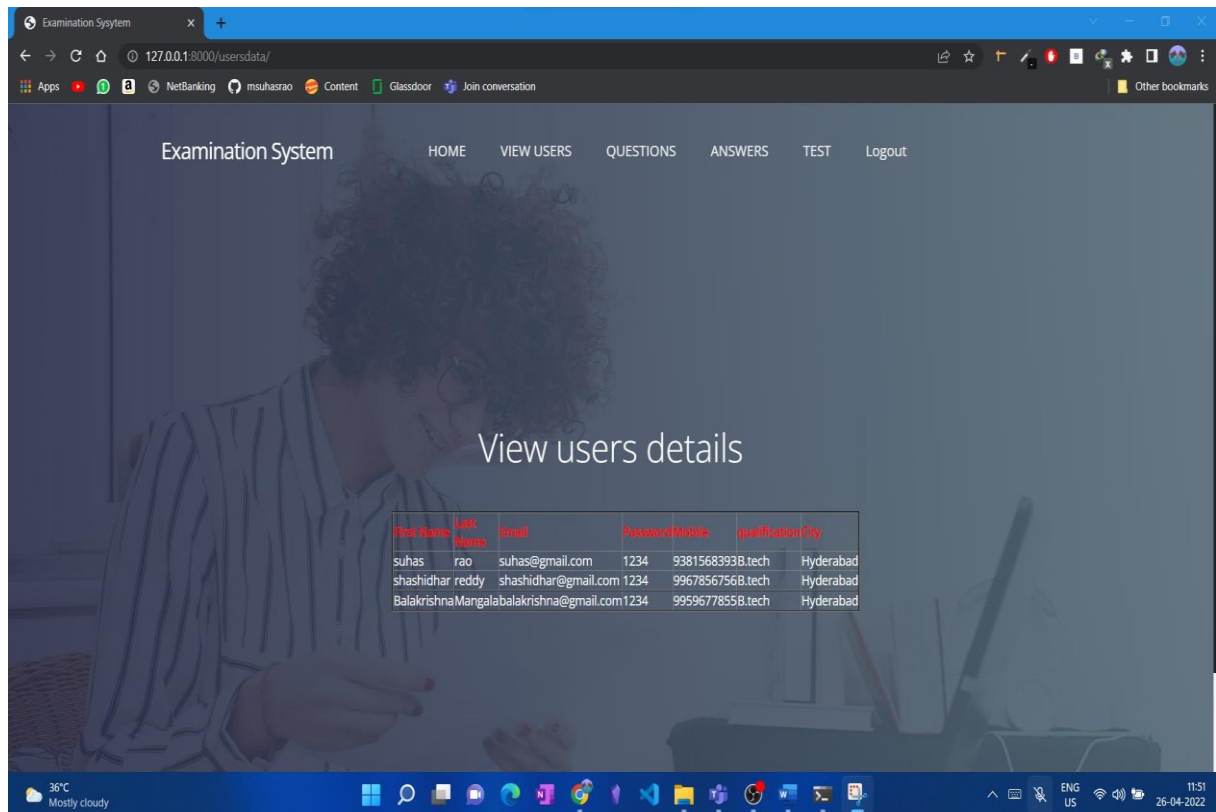**It consists of 4 – view users, questions, answers and test, logout.**

Fig 7.4 View Users page
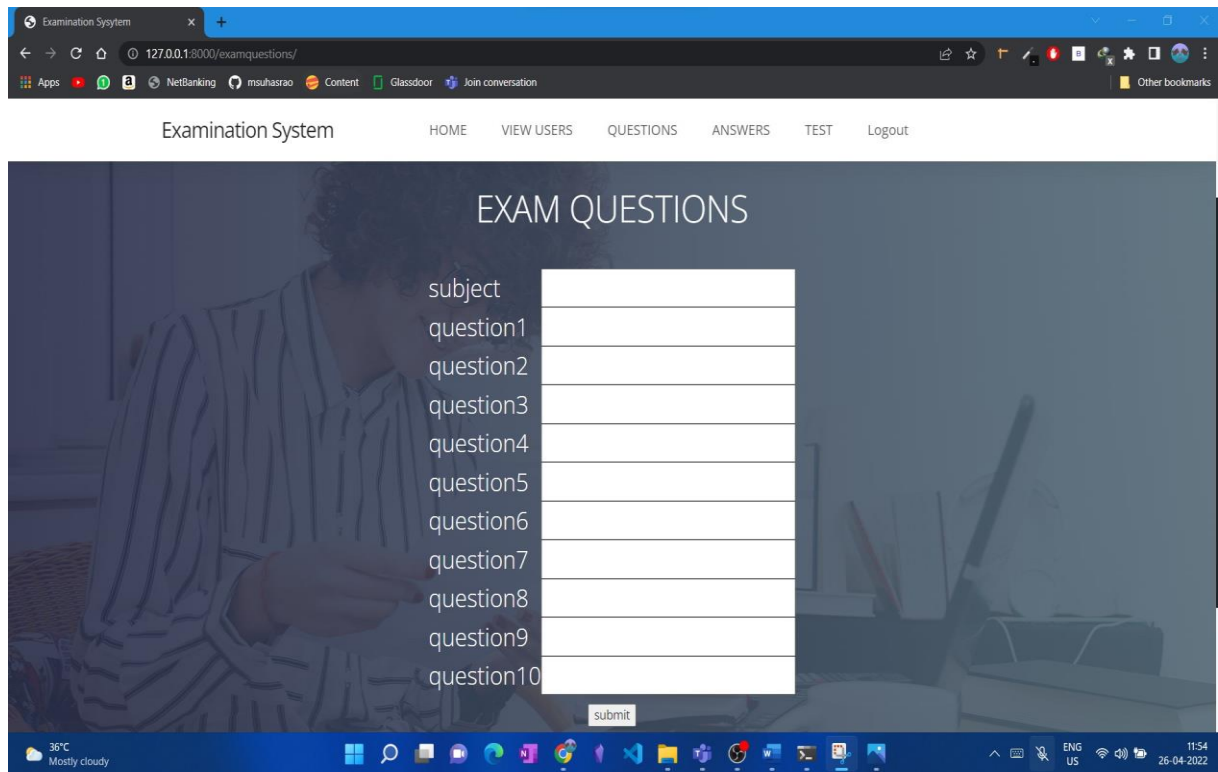
It displays the details of all the examinees

Fig 7.5 Exam Questions

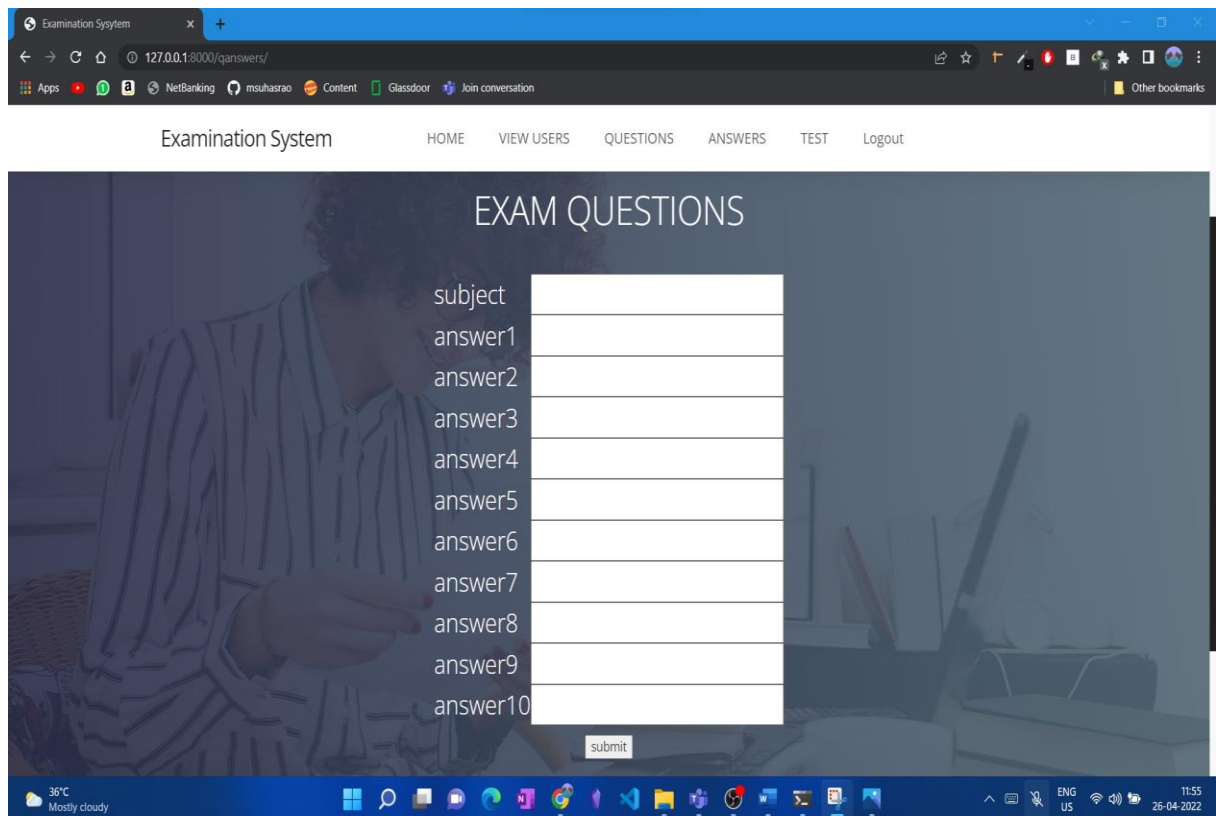**It gives the access to upload the exam questions for a particular subject**

Fig 7.6 Exam Answers page

It gives the access to upload the exam answers for a particular subject
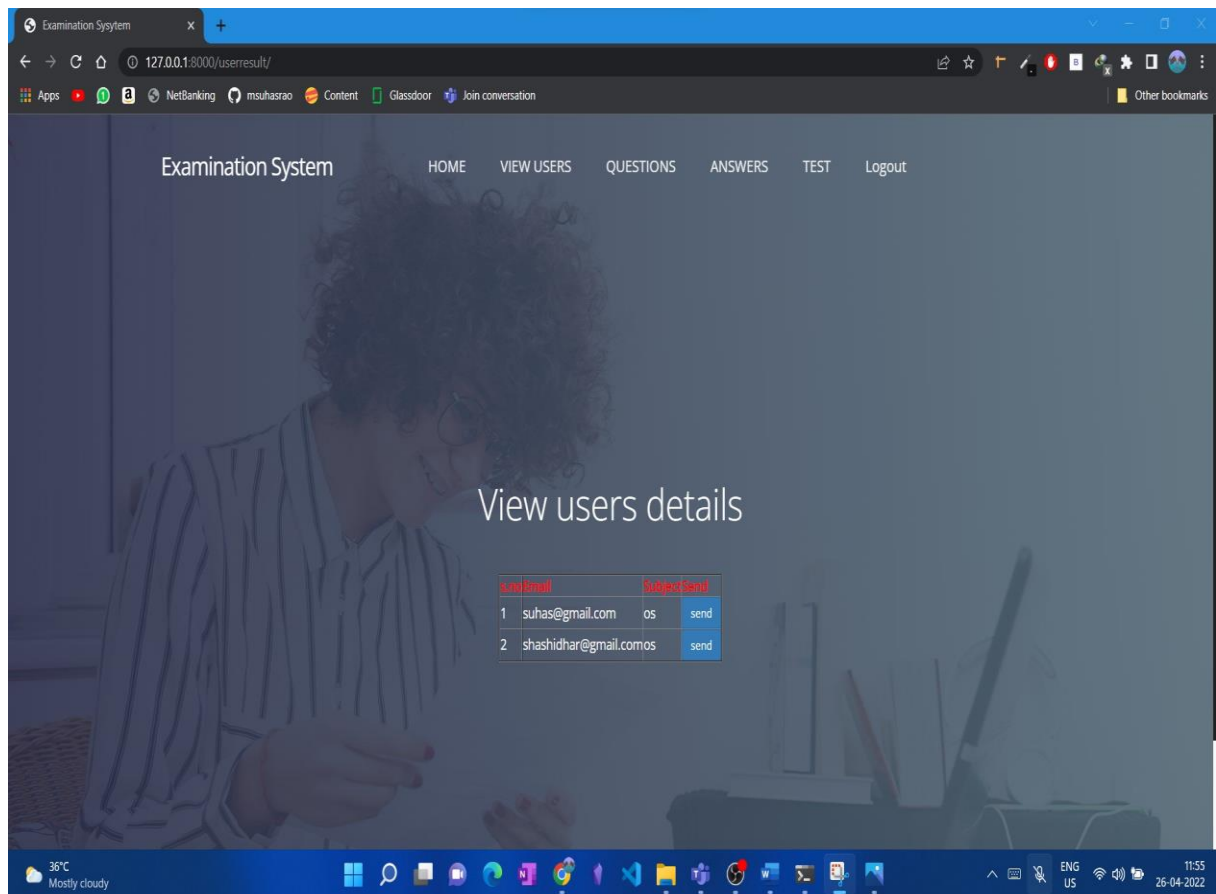
Fig 7.7 Test result page

It gives the examiner to release the result of the students for the particular subject which they have attempted
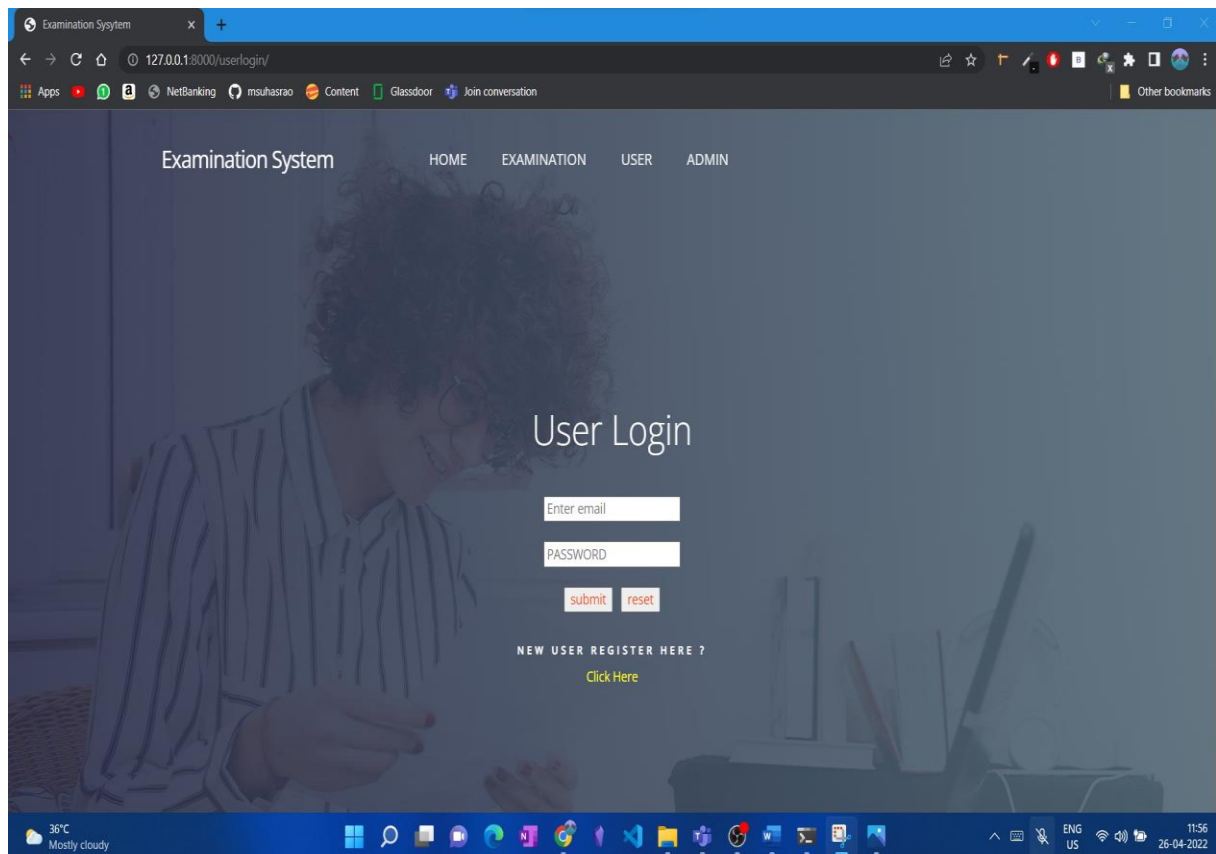
Fig 7.8 User login page

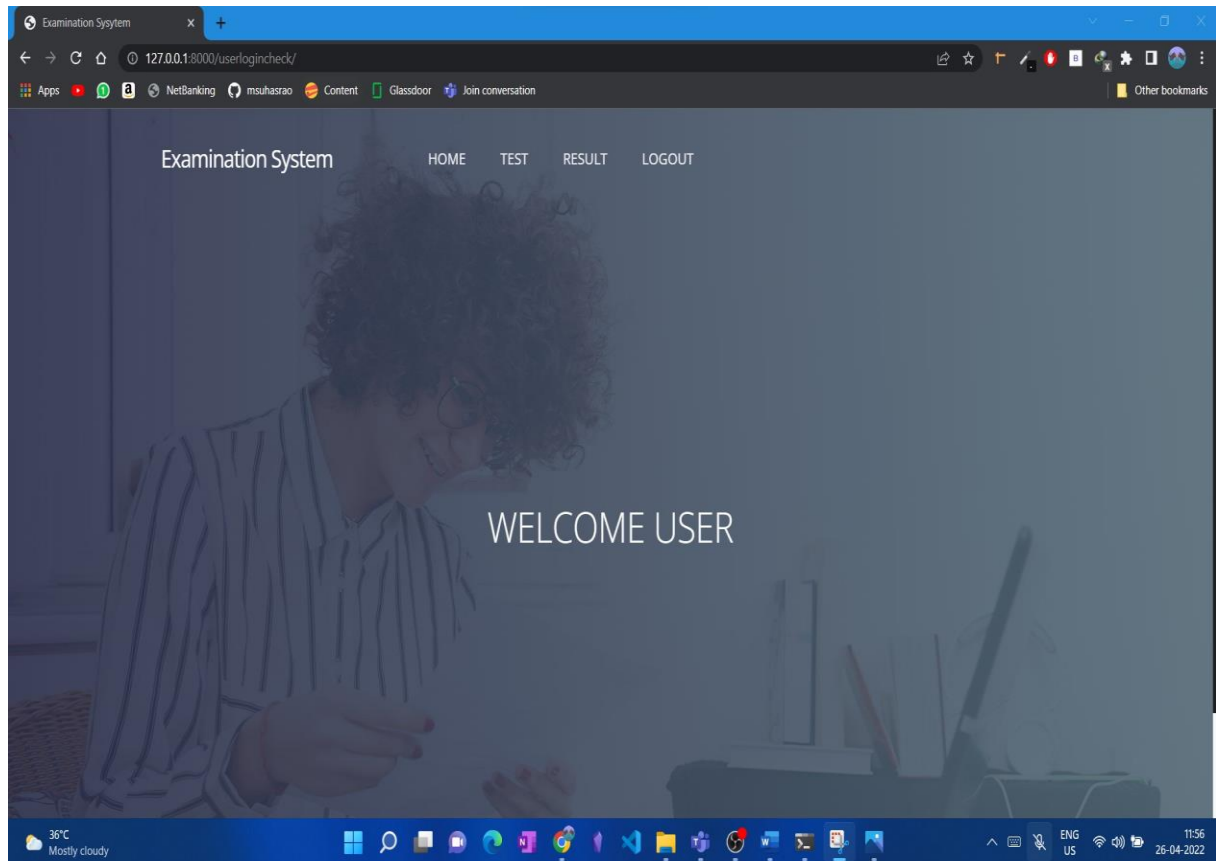**It helps the user to login to the portal**

Fig 7.9 Home page of User

It gives the access to the portal of the user to write exams
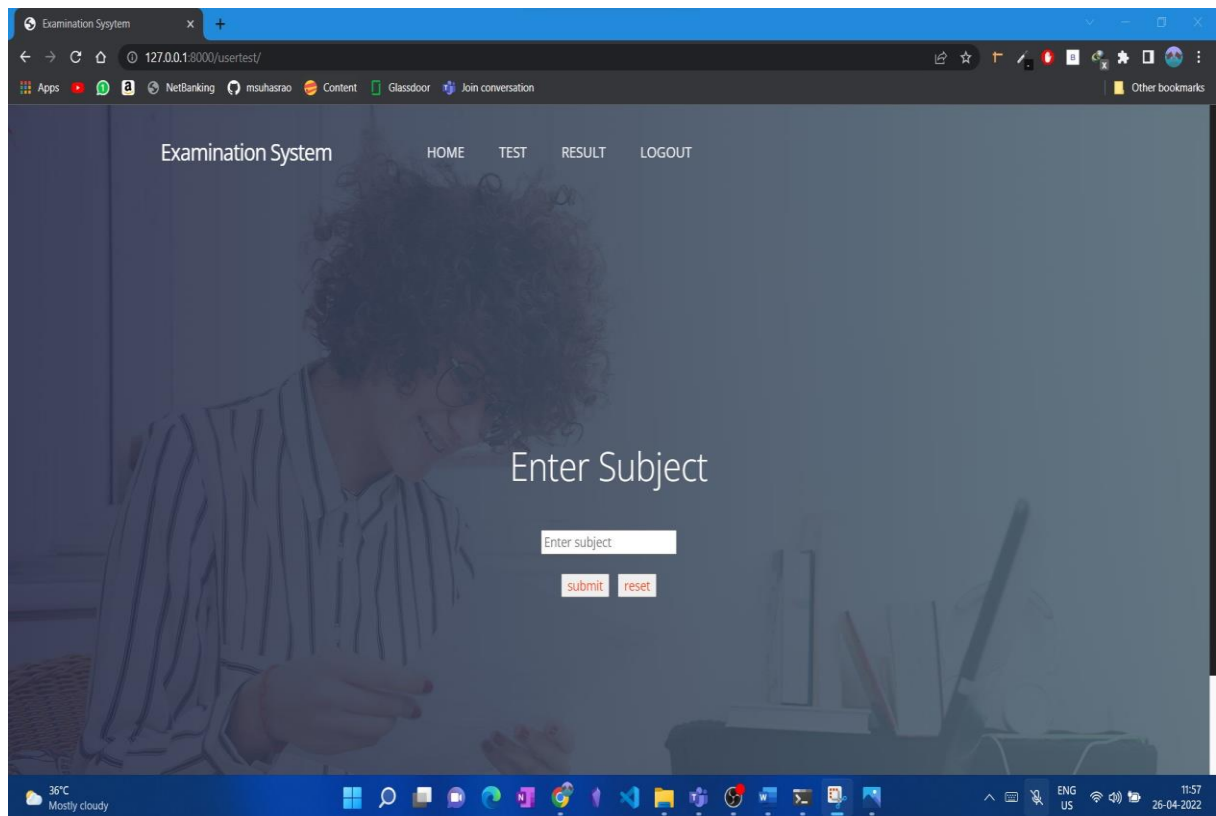
Fig 7.10 Choose test Subject page

It gives the user access to the test by a particular subject
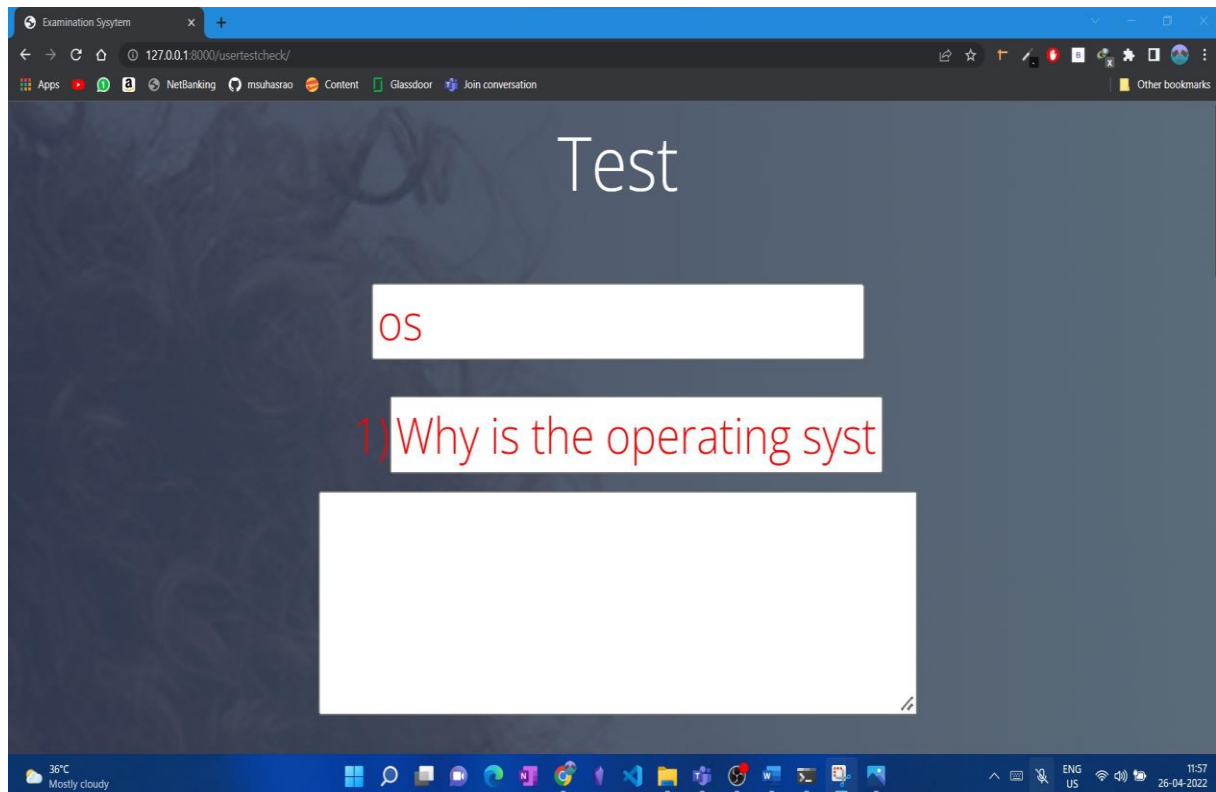
Fig 7.11 Test interface page

It displays the Test interface of the student where students can write answers to the subject
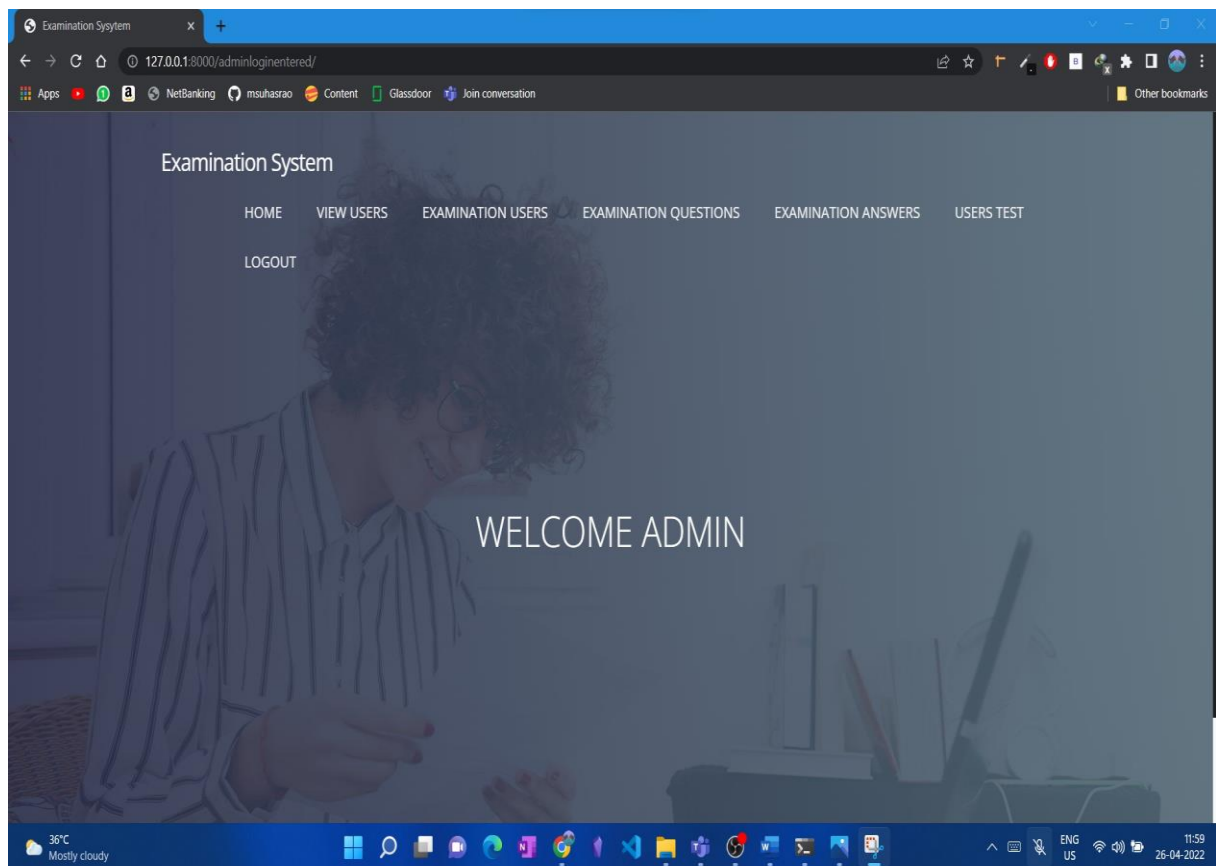
Fig 7.12 Admin Login

It gives details of view users, examination users, examination questions and answers , users test

Fig 7.13 User details

Admin can view the users who are registered in the portal and activate the genuine users

Fig 7.14

Admin can view the examinee who registered into the portal and can activate the genuine examination users

Examination System

HOME   VIEW USERS   EXAMINATION USERS   EXAMINATION QUESTIONS   EXAMINATION ANSWERS   USERS TEST

LOGOUT

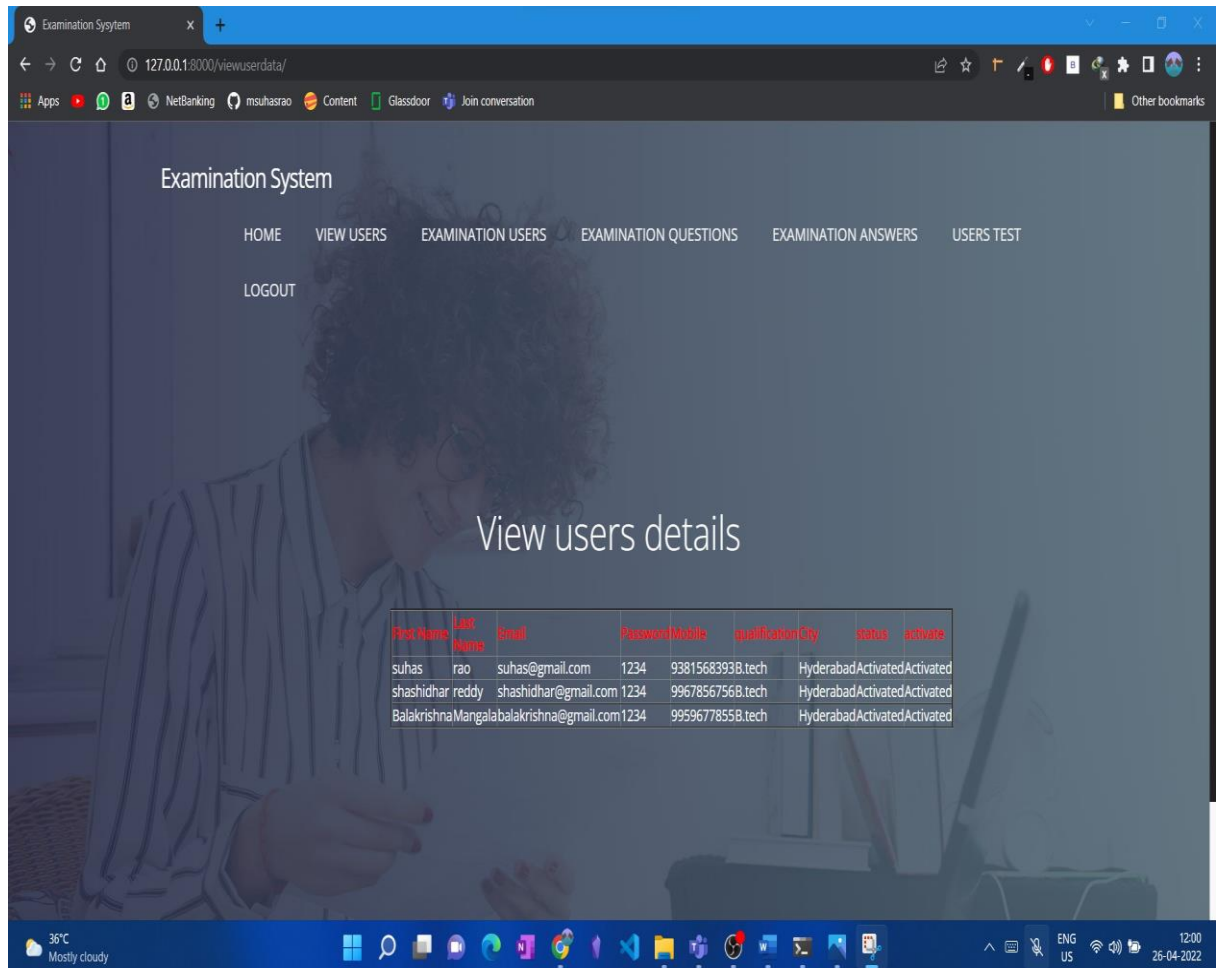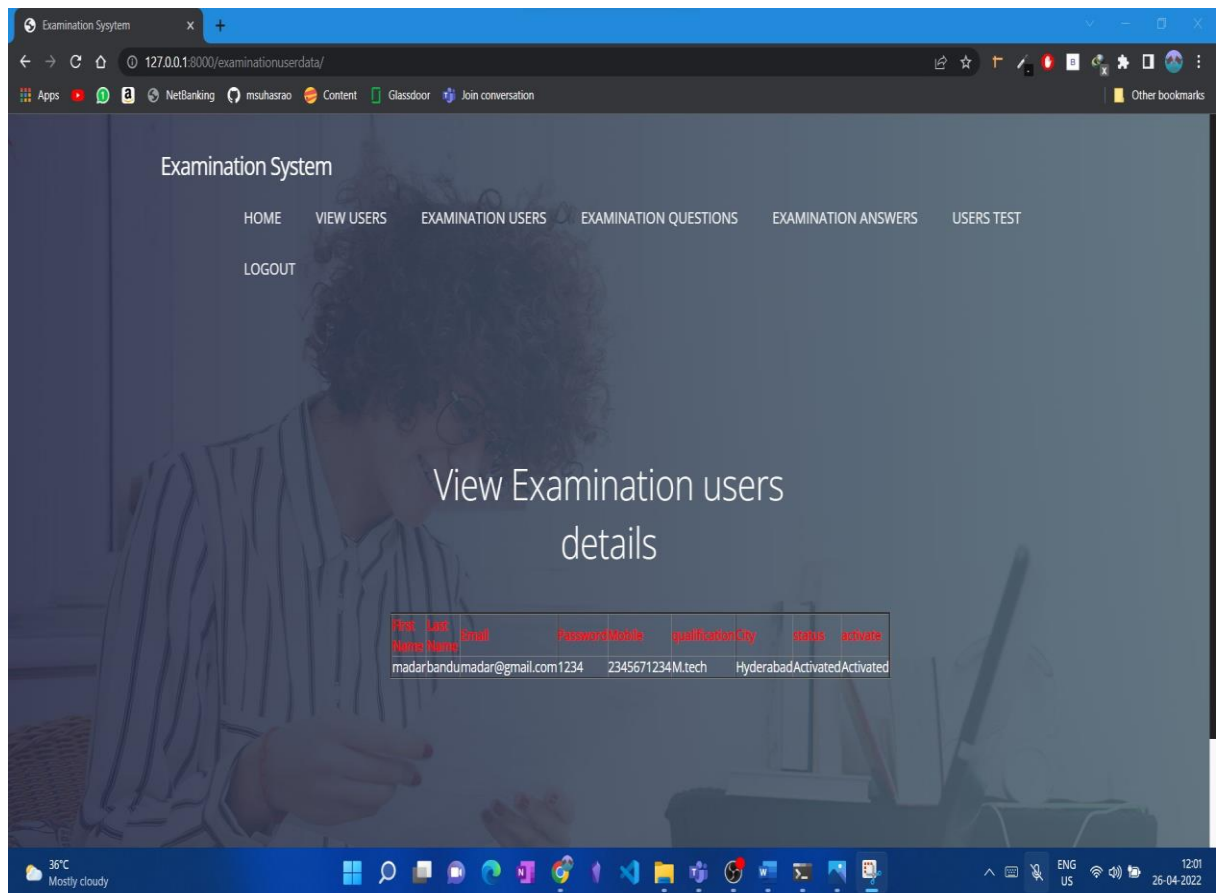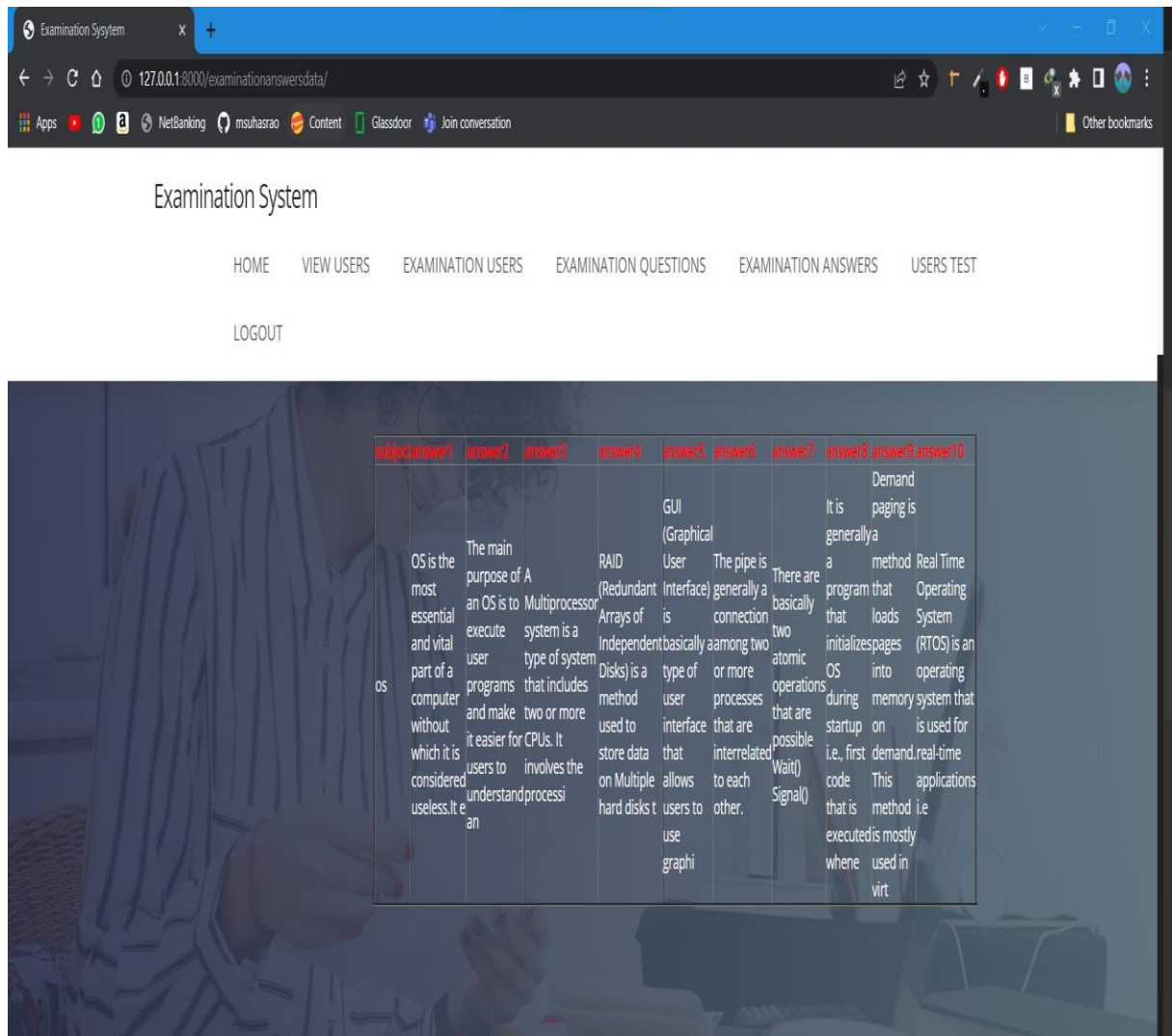| subject	answer1	answer2	answer3	answer4	answer5	answer6	answer7	answer8	answer9	answer10
os	OS is the most essential and vital part of a computer without which it is considered useless.It e	The main purpose of an OS is to execute user programs and make it easier for users to understand an	A Multiprocessor system is a type of system that includes two or more CPUs. It involves the processi	RAID (Redundant Arrays of Independent Disks) is a method used to store data on Multiple hard disks t	GUI (Graphical User Interface) is basically type of user interface that allows users to use graphi	The pipe is generally a connection among two or more processes that are interrelated to each other.	There are basically two atomic operations that are possible Wait() Signal()	It is generally a program that initializes OS during startup i.e., first code that is executed when e	Demand paging is a method that loads pages into memory on demand. This method is mostly used in virt	Real Time Operating System (RTOS) is an operating system that is used for real-time applications i.e

Fig 7.15

It shows the questions and answers uploaded by the examination user and can be viewed by the admin
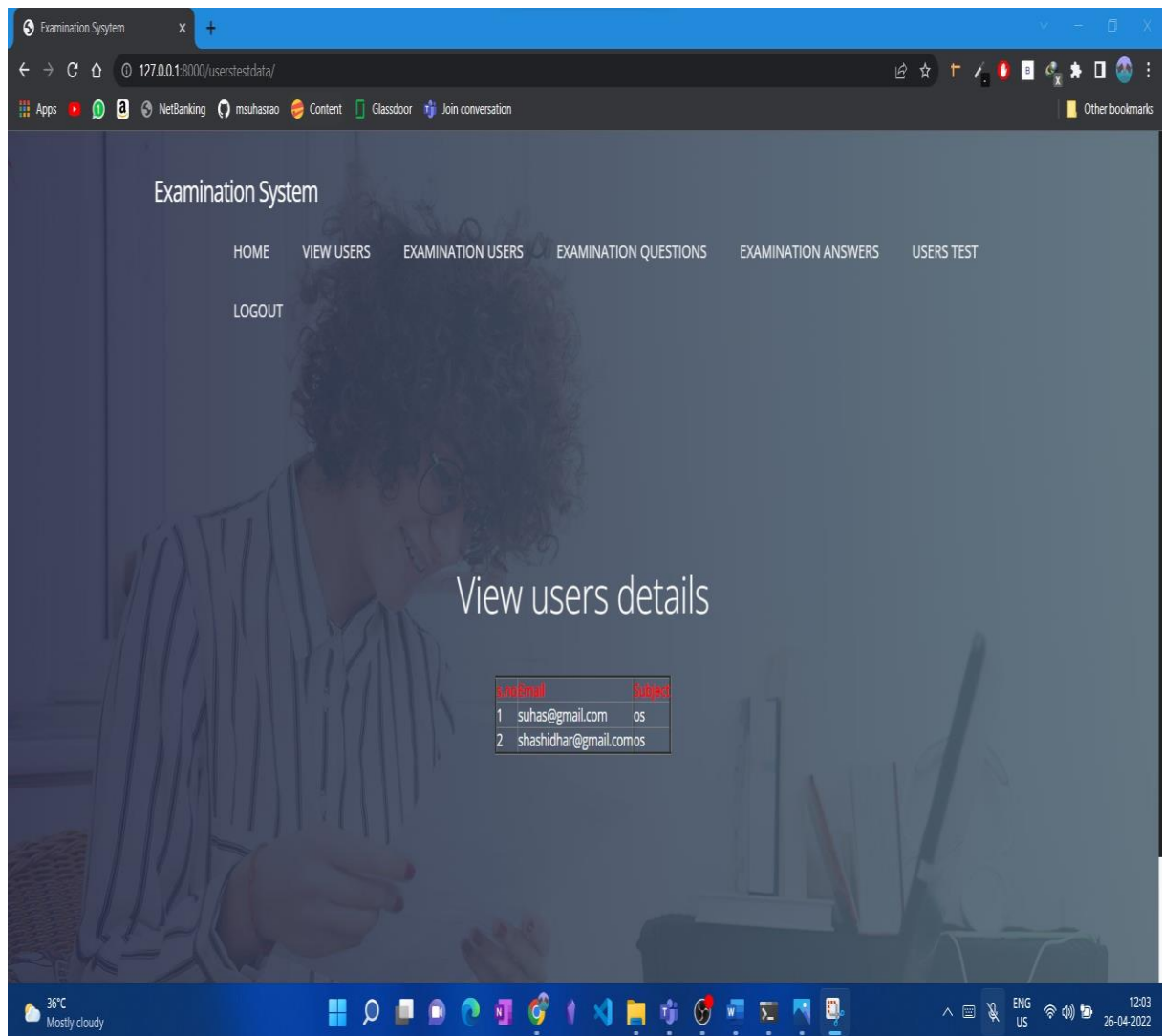
Fig 7.16

It shows the admin about the different users who have taken the test in the portal and for which subject the test has been submitted.

8. Conclusion

In conclusion, It can be seen by conducting tests using such an algorithm at regular intervals that one can determine the trend in the marks obtained by different students and we can give them an analyzed report on the different subjects they need to focus on for which they are weak. With the existing data, we can also implement a predictive machine learning model on the data so that it can predict marks that the students will score in the future. It is observed that students mainly study those subjects that are placement oriented or which are required for placement purpose only. While students neglect the subjects of their core domain. Deep knowledge in the domain is required as it is of no use to study if you do not have a core domain knowledge. So it can help students get quality knowledge as everything will be digital and there will be no cumbersome process of conducting a pen-paper test. Also, answers are evaluated at that moment itself and the student can see the solutions and can correct the mistakes or errors committed while appearing for the exam.

9. Future Enhancements

➢ Online Examination System is widely used as compared to other exams.

➢ Every software may have some cases of bugs, errors, security related problems or system faults.

➢ There are many problems or system faults for example; computer collapse or crashes due to power supply problem will invalidate efforts of number of students.

➢ These bugs must be identified and solved for improving quality of software. So, in future we can develop more secure software by using advanced technologies

➢ By introducing new security systems using biometrics, we can identify the student's true identity by analyzing digital signature or by finger print or by capturing images and also by using online live proctoring.

10. Bibliography

[1] K. Jayakodi, M. Bhandara and I. Perera "An automatic classifier for exam questions in Engineering: A process for Bloom's taxonomy", IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), (2015)

[2] N. Ishikawa, K. Umemoto, Y. Watanabe, Y. Okada, R. Nishimura and M. Murata "Detection of users suspected of using multiple user accounts and manipulating evaluations in a community site", IEEE Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering, (2010)

[3] B. Kaur, and S. Jain "Keyword extraction using machine learning approaches", IEEE 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall), (2017)

[4] R. P. Futrelle, J. Satterley, and T. McCormack "NLP-NG — A new NLP system for biomedical text analysis", IEEE International Conference on Bioinformatics and Biomedicine Workshop, (2009)

[5] M. Revathy, and M. L. Madhavu "Efficient author community generation on Nlp based relevance feature detection", IEEE International Conference on Circuit ,Power and Computing Technologies (ICCPCT), (2017)

[6] W. Nei, Y. Wu, D. Hu, L. Wang, and Y. Li "Data Management and Analysis of Intelligent Examination Scoring System of Simulation Training System", IEEE 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, (2013)

[7] G. Zhang, and H. Ke "Design of Paperless Examination System for Principles of Database Systems", IEEE International Conference on Research Challenges in Computer Science, (2009)