**An Industry-Oriented Major Project Report**

**On**

# “ShareNest: A Collaborative Academic Project Sharing Platform for College Communities”

**Submitted in Partial Fulfillment of the Academic Requirement for the Award of Degree**

# BACHELOR OF TECHNOLOGY

**in**

### Computer Science and Engineering

**Submitted by:**

|  |  |
| --- | --- |
| **ANKITH REDDY MOMULA** | **23R01A05EL** |
| **NIVAS MULLURI** | **23R01A05EM** |
| **RAJASHEKAR MUPPIDI** | **23R01A05EN** |
|  |  |

### Under the esteemed guidance of Mrs. S. Sresta Joshi

(Associate Professor, CSE Department)



# CMR INSTITUTE OF TECHNOLOGY

#### (UGC AUTONOMOUS)

**Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC ‘A+’**

**Kandlakoya (V), Medchal Dist – 501 401** [**www.cmritonline.ac.in**](http://www.cmritonline.ac.in/)

# 2024-2025

**CMR INSTITUTE OF TECHNOLOGY**

#### (UGC AUTONOMOUS)

**Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC ‘A+’**

**Kandlakoya (V), Medchal Dist – 501 401** [**www.cmritonline.ac.in**](http://www.cmritonline.ac.in/)

****

**CERTIFICATE**

This is to certify that an industry oriented Major Project entitled with “ **ShareNest: A Collaborative Academic Project Sharing Platform for College Communities** ” is being

|  |  |
| --- | --- |
| submitted by |  |
| **ANKITH REDDY MOMULA** | **23R01A05EL** |
| **NIVAS MULLURI** | **23R01A05EM** |
| **RAJASHEKAR MUPPIDI** | **23R01A05EN** |
|  |  |

To JNTUH, Hyderabad in partial fulfillment of the requirement for award of the degree of B- Tech in CSE and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University for award of any other degree or diploma.

|  |  |  |
| --- | --- | --- |
| **Signature of Guide** | **Signature of Coordinator** | **Signature of HOD** |

# ACKNOWLEDGEMENT

We are extremely grateful to Director, Principal and Head of Department, Dept of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during the entire duration.

We are extremely thankful to Major Project Coordinator and internal guide Dept of Computer Science and Engineering, CMR Institute of Technology for their constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided us directly or indirectly for every step towards success.

|  |  |
| --- | --- |
| **ANKITH REDDY MOMULA** | **23R01A05EL** |
| **NIVAS MULLURI** | **23R01A05EM** |
| **RAJASHEKAR MUPPIDI** | **23R01A05EN** |
|  |  |

iii

# ABSTRACT

The objective of this project is to design and develop **ShareNest**, an interactive and user-friendly web platform tailored for a college community. The platform aims to provide students with a centralized space where they can **upload, share, and rate academic projects** across various subjects. By facilitating seamless project sharing, **ShareNest** fosters a culture of **collaboration and innovation**, allowing students to learn from each other’s work and gain inspiration for their own projects.

This system serves as a **valuable repository of knowledge**, benefiting both current students and future batches by preserving high-quality academic contributions. Users will be able to **browse projects based on categories, popularity, or recent uploads**, ensuring easy access to relevant information. Additionally, the platform will incorporate **a rating and feedback system**, enabling students to evaluate and discuss shared projects, further enhancing their educational value.

By promoting **peer learning and academic engagement**, **ShareNest** encourages students to refine their skills and contribute meaningfully to the community. The platform will also feature **a user-friendly interface, robust search functionality, and secure user authentication** to ensure a seamless and secure experience. Ultimately, **ShareNest** will serve as an essential academic hub, driving **knowledge-sharing and continuous learning** within the college environment.

iv

|  |  |
| --- | --- |
| **ACKNOWLEDGEMENT ABSTRACT**  [**INDEX**](#_bookmark0)  **LIST OF FIGURES** | iii iv [v](#_bookmark0)  vi |
| **1. INTRODUCTION** | **7** |
| **2. SYSTEM ANALASYS** | **8** |
| **2.1 Existing System** | **8** |
| **2.2 Disadvantages of Existing System** | **8** |
| **2.3 Proposed System** | **9** |
| **2.4 Advantages of Proposed System** | **9** |
| **3. SYSTEM STUDY** | **10** |
| **3.1 Feasibility Study** | **10** |
| **3.1.1 Economic Feasibility** | **10** |
| **3.1.2 Technical Feasibility** | **10** |
| **3.1.3 Social Feasibility** | **11** |
| **4. HARDWARE AND SOFTWARE REQUIREMENTS** | **12** |
| **4.1 Software Requirements** | **12** |
| **4.2 Hardware Requirements** | **12** |
| **5. LITERATURE REVIEW** | **13** |
| **6. PROJECT CATEGORY** | **17** |
| **6.1 Data flow Diagrams** | **18** |
| **6.2 Use case diagrams** | **19** |
| **6.3 Class Diagrams** | **20** |
| **6.4 Sequence Diagrams** | **21** |
| **6.5 Flow Chart Diagrams** | **22** |
| **7. MODULES** | **24** |
| **8. IMPLEMENTATION** | **25** |
| **8.1 Algorithms** | **30** |
| **8.2 Source Code** | **47** |
| **9. SCREEN SHOTS** | **48** |
| **10. TESTING** | **60** |
| **11. CONCLUSION** | **62** |
| **12. REFERENCES** | **63** |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Particulars** | **Page No.** |
| **5.1** | **Architecture** | **16** |
| **7.1** | **Dataflow Diagram** | **18** |
| **7.2** | **Use Case Diagram** | 19 |
| **7.3** | **Class Diagram** | 20 |
| **7.4.1** | **Sequence Diagram** | 21 |
| **7.5.1** | **Remote user Flowchart** | 22 |
| **7.5.2** | **Service Provider Flowchart** | 23 |
| **9.1** | **Home Page** | 47 |
| **9.2** | **Service Provide Login Page** | 48 |
| **9.3** | **Remote User Login Page** | 48 |
| **9.3.1** | **Remote User Registration Page** | 49 |
| **9.3.2** | **Remote User Prediction Page** | **49** |
| **9.3.3** | **List of Remote Users** | **50** |
| **9.3.4** | **User Prediction Ratio** | **50** |
| **9.3.5** | **User Prediction Ratio in Pie Chart** | **58** |
| **9.4** | **Accuracy According to Algorithms** | **58** |
| **9.5** | **Bar chart for accuracy according to algorithms** | **59** |

vi

# INTRODUCTION

Welcome to **ShareNest**, a dynamic and interactive web platform designed to revolutionize academic collaboration within the college community. As the **Project Head**, I am thrilled to introduce this initiative aimed at creating a **centralized repository** where students can **upload, share, and rate academic projects** across various subjects. This platform provides an opportunity for students to showcase their work, gain recognition, and inspire others by sharing innovative academic contributions.

ShareNest fosters a **culture of collaboration, innovation, and engagement**, allowing students to explore high-quality projects, provide constructive feedback, and recognize outstanding work through a structured **rating and review system**. With an intuitive interface, powerful search functionality, and secure user authentication, the platform ensures seamless access to academic resources while maintaining data privacy and integrity.

Beyond just a storage hub, ShareNest acts as a **learning ecosystem** where students can draw insights from past projects, refine their skills, and improve the quality of their own work. By encouraging **peer learning and mentorship**, the platform promotes intellectual growth and motivates students to push their academic boundaries. Faculty members can also utilize ShareNest to track student progress, identify academic trends, and encourage interdisciplinary collaborations, further enriching the educational experience.

To enhance user engagement, ShareNest incorporates **advanced filtering options**, enabling students to browse projects based on categories, ratings, and popularity, ensuring easy access to relevant academic content. The platform also introduces a **discussion forum**, where students can ask questions, seek guidance, and exchange ideas with peers and mentors, creating a thriving academic community.

By bridging the gap between past and present student contributions, ShareNest serves as an invaluable academic hub that **promotes knowledge sharing, peer-driven education, and continuous learning**. Through this initiative, we aim to empower students, foster creativity, and cultivate a culture of academic excellence, ensuring that knowledge is preserved and passed down to future generations of learners.

# SYSTEM ANALASYS

# Existing System

# In the current scenario, students primarily share academic projects through informal channels such as social media groups, messaging apps, or personal interactions. These methods lack organization, accessibility, and long-term availability, making it difficult for students to find relevant projects or refer to past work for learning purposes. Additionally, many projects are stored on personal devices or local college servers, restricting access to a limited audience.

# There is no centralized repository where students can systematically upload, browse, and evaluate academic projects across various subjects. The absence of a structured rating or feedback system prevents students from assessing the quality of shared work, limiting constructive peer evaluation. Moreover, searching for specific projects is time-consuming and inefficient, as there are no proper categorization or filtering options.

# Furthermore, students often face difficulty in collaborating with peers from different departments, as there is no dedicated platform to encourage interdisciplinary knowledge exchange. Faculty members also struggle to track and preserve noteworthy student contributions, resulting in the loss of valuable academic resources over time. The lack of a secure and userfriendly digital platform hinders students from efficiently sharing, discovering, and building upon past academic work, ultimately reducing opportunities for collective learning and innovation.

# DISADVANTAGES OF EXISTING SYTEM:

#  Lack of Centralized Storage Academic projects are scattered across different platforms, making it difficult for students to access past work efficiently.

 **No Proper Evaluation System**  
There is no structured **rating or feedback mechanism**, preventing students from assessing project quality and learning from constructive criticism.

 **Limited Accessibility & Collaboration**  
Projects are often stored on personal devices or shared in small groups, restricting access to a wider academic community and limiting interdisciplinary collaboration.

 **Inefficient Searching & Organization**  
Without proper categorization or filtering options, finding relevant academic projects is time-consuming and unorganized.

 **Risk of Data Loss & Duplication**  
Valuable student projects may get lost or forgotten due to lack of a structured repository, and duplication of efforts occurs as students unknowingly work on similar projects without reference to past work.

## PROPOSED SYSTEM:

## ShareNest: The Solution A dedicated web platform to address the challenges of project sharing and collaboration. Key Features:Project Upload and Sharing: Easy-to-use interface for submitting and sharing projects. Subject-Based Categorization: Projects organized by subject and course for easy browsing. Rating and Review System: Students can rate and provide feedback on projects. Collaboration Tools: Integrated features for connecting with other students and forming project teams. Search Functionality: Robust search capabilities to quickly find relevant projects. User Profiles: Showcase your projects and skills to the community.

#### ADVANTAGES OF PROPOSED SYSTEM:

#### Improved Collaboration and Knowledge Sharing The platform enables students to share and access a diverse range of academic projects across various subjects, promoting better collaboration and knowledge exchange among peers.

1. **Comprehensive Project Analysis and Feedback**ShareNest’s built-in rating and feedback system ensures that projects are evaluated from multiple perspectives, offering students valuable insights and constructive criticism to improve their work.
2. **Real-time Updates and Continuous Access**Students can continuously access and review newly uploaded projects, keeping the platform dynamic and up-to-date with the latest contributions from the college community.
3. **Enhanced Search and Categorization**The advanced search and filter capabilities of ShareNest allow students to find relevant projects quickly by subject, topic, or rating, improving the overall user experience and efficiency.
4. **Encouragement of Healthy Competition**The leaderboard feature fosters healthy competition among students by showcasing the highest-rated projects, motivating them to strive for excellence in their academic work.
5. **Objective and Transparent Evaluation**By automating the rating system and ensuring peer-reviewed feedback, ShareNest reduces subjectivity, making project evaluations more objective and transparent for students and instructors alike.

**7. SCOPE OF THE PROJECT**

## The ShareNest platform is designed to provide a centralized, interactive, and user-friendly solution for students to upload, share, and rate academic projects across various subjects. It aims to bridge the gap between past and present student contributions, creating a sustainable repository of academic resources that will benefit both current students and future batches.

## This system will allow users to browse projects based on categories, ratings, and relevance, ensuring easy access to quality academic content. A rating and feedback system will be integrated to help students evaluate project quality and engage in peer-to-peer learning. To enhance collaboration, the platform will also feature a discussion forum, where students can ask questions, share insights, and seek guidance from peers and faculty.

## The platform will incorporate secure user authentication, ensuring that only verified students and faculty members can contribute and access academic resources. Additionally, faculty members can utilize the system to monitor academic trends, guide students, and promote interdisciplinary collaboration.

## ShareNest is designed to be scalable and adaptable, with the potential to expand beyond project sharing to include research papers, study materials, and interactive learning resources. With its intuitive interface, advanced search capabilities, and structured data management, ShareNest will serve as a one-stop academic hub, promoting knowledge-sharing, innovation, and continuous learning within the college community.

**3.SYSTEM STUDY**

#### 3.1 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**

**3.1.1 ECONOMICAL FEASIBILITY**

The development and implementation of **ShareNest** must be cost-effective while providing maximum value to the college community. The project requires **minimal investment** as it primarily involves **web development, hosting, and maintenance costs**. Open-source technologies and cloud-based storage solutions can be utilized to **reduce expenses**. The long-term benefits of **knowledge preservation, improved academic collaboration, and reduced duplication of efforts** outweigh the initial development costs, making the project **economically viable**.

### 3.1.2 TECHNICAL FEASIBILITY

The project is technically feasible as it leverages **widely available web development technologies** such as **HTML, CSS, JavaScript, PHP, and databases like MySQL or Firebase**. Modern frameworks and cloud computing can ensure **scalability, security, and smooth performance**. User authentication, file management, and data retrieval mechanisms will be implemented using **reliable and well-established technologies**. Given the availability of skilled developers and existing technological infrastructure within the college, the technical implementation of **ShareNest** is **achievable**.

**3.1.3 SOCIAL FEASIBILITY**

The success of **ShareNest** depends on its acceptance and usability within the college community. The platform is designed to **enhance student engagement, encourage knowledge sharing, and foster collaboration** among peers. By providing an **interactive and user-friendly interface**, students will be more inclined to upload and explore projects. Faculty members can also benefit from tracking academic trends and mentoring students effectively. As the platform **aligns with the needs and interests** of students and faculty, it is **socially feasible and highly beneficial** for the college environment.

# REQUIREMENT ANALYSIS

## The ShareNest platform requires a secure and scalable web-based system that allows students to upload, share, and rate academic projects. It must include user authentication, project categorization, search functionality, and a feedback system to ensure accessibility and engagement. The platform should be developed using modern web technologies with a responsive and intuitive interface for seamless navigation.

## 4.1. SOFTWARE REQUIREMENTS:

For developing the application the following are the Software Requirements:

Operating system : Windows 7 Ultimate.

Coding Language : Python.

Front-End : HTML, CSS, JavaScript (React.js for an interactive UI)

Designing : Html,css,javascript , Bootstrap/Tailwind CSS for responsive design

Data Base : MySQL , or Firebase for secure data storage and retrieval.

**OPERATING SYSTEMS SUPPORTED:**

Windows 7

Windows XP

Windows 8

**TECHNOLOGIES AND LANGUAGES USED TO DEVELOP:**

1.Python, HTML, CSS, JavaScript (React.js) and mySQL.

* + 1. **DEBUGGER AND ENMULATOR:**

Any Browser (Particularly Chrome)

* 1. **HARDWARE REQUIREMENTS:**

For developing the application the following are the Hardware Requirements:

System : Pentium IV 2.4 GHz.

Hard Disk : 40 GB.

Floppy Drive : 1.44 Mb.

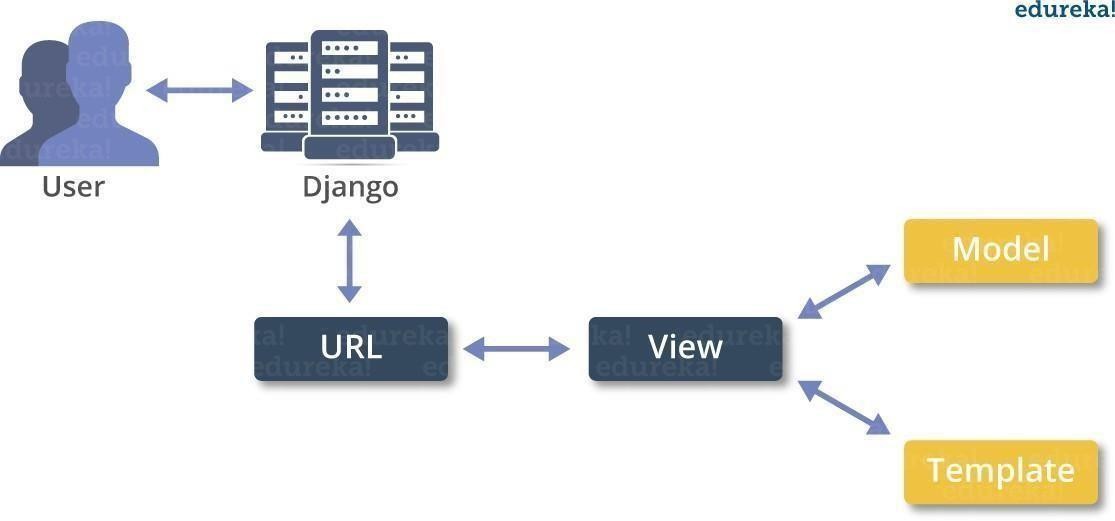
Monitor : 14’ Colour Monitor.

Mouse : Optical Mouse.

Ram : 512 Mb.

# 

## 



**5. LITERATURE REVIEW**

# The concept of online academic collaboration platforms has evolved significantly over the years, driven by advancements in web technologies, cloud computing, and open-source learning models. Platforms such as GitHub, ResearchGate, and Google Drive have provided students and researchers with tools to share, collaborate, and improve academic projects. However, these platforms do not cater specifically to the structured needs of college students looking for a centralized repository of academic projects, which is where ShareNest aims to bridge the gap.

# Existing Research on Knowledge-Sharing Platforms

# Various studies emphasize the importance of peer learning and knowledge sharing in an academic environment. Research has shown that collaborative learning platforms enhance student engagement, promote innovation, and facilitate better knowledge retention. Several existing platforms, such as Coursera and Edmodo, allow students to access learning materials but lack an integrated project-sharing mechanism that fosters active contribution and feedback.

# Limitations in Current Project-Sharing Methods

# Lack of Centralization: Students often rely on fragmented sources such as social media groups, cloud drives, and personal blogs, leading to difficulty in accessing relevant academic projects.

# Limited Peer Feedback: Most platforms do not provide structured rating and review mechanisms, which can help students refine and improve their projects.

# Inefficient Project Discovery: Without an organized categorization system, searching for projects relevant to a particular subject or domain becomes challenging.

# No Institutional Support: Many institutions lack dedicated digital repositories, making it hard for future batches to access previous academic work for reference and innovation.

# Technological Advancements Enabling ShareNest

# With the rise of cloud storage, real-time collaboration, and AI-driven search algorithms, modern web platforms can now efficiently manage, store, and recommend projects to users. Technologies such as Python (Django/Flask), MySQL databases.

**6. INPUT AND OUTPUT DESIGN**

**INPUT DESIGN**

The input design of **ShareNest** plays a crucial role in ensuring an efficient, user-friendly interaction between students, faculty, and the system. It defines how users enter data, ensuring accuracy, security, and ease of use. The input mechanisms include **student and faculty registrations, project uploads, rating and review submissions, and search queries**. To **minimize errors and maintain data integrity**, various validation checks are implemented, such as ensuring valid email formats, restricting duplicate project entries, and preventing unauthorized access.

* Ensures a seamless interaction between users and the ShareNest platform.
* Users provide inputs like **registration details, project uploads, ratings, and comments**.
* Input fields are structured to **minimize errors, prevent redundancy, and enhance security**.
* Includes validation checks for **file formats, required fields, and user authentication**.

**OBJECTIVES**

* Efficient Knowledge Sharing Facilitate a seamless and organized platform where students and faculty can upload, categorize, and access academic projects, research papers, and related resources.
* User Engagement and Collaboration Provide interactive features such as ratings, reviews, and discussions to enhance collaboration and encourage meaningful academic interactions among users.
* Secure and Accessible Data Management Ensure secure data storage, role-based access control, and easy retrieval of information while maintaining data integrity and privacy.

**OUTPUT DESIGN**

The output design of **ShareNest** is structured to deliver **clear, well-organized, and meaningful information to its users**. The platform generates multiple types of outputs, such as **project listings, search results, user reviews, ratings, administrative reports, and notifications**. These outputs ensure that students and faculty can easily retrieve and interact with relevant academic content.

The system is designed to **present data in a visually appealing and structured format**, ensuring that information is easily understandable. Projects are displayed with **detailed descriptions, author details, categories, and ratings** to help users make informed decisions. Search results are sorted based on **relevance, popularity, or date** to improve accessibility.

The output form of an information system should accomplish one or more of the following objectives.

 Generates structured **project listings, user profiles, ratings, and admin reports**.

 Displays search results, project details, and community feedback in a **well-organized manner**.

 Admins receive **dashboard reports** on user activity and project approvals.

 Provides **real-time notifications** for approvals, comments, and updates.

**7.MODULES**

* User Management.
* Project Management.
* Rating & Review.
* Search.
* Admin Modules.
* Future Modules

**MODULES DESCRIPTION:**

**User Management:**

**This module handles user registration, login, profile management, and authentication. It ensures secure access through role-based permissions, distinguishing between students, admins, and contributors.**

**Project Management:**

**Users can upload academic projects, categorize them based on subjects, and manage their listings. This module allows editing, deleting, and viewing project details, ensuring a structured repository.**

**Rating & Review:**

# Users can submit ratings and comments on uploaded projects to provide feedback. A moderation system ensures that inappropriate or irrelevant reviews are filtered out to maintain content quality.

# Search:

# Users can search for projects using keywords, filters, and advanced options like category or user-based searches. This module ensures quick access to relevant academic projects.

# Admin Modules:

# Admins manage project categories, oversee user activities, and generate reports on platform usage. They also approve or reject uploaded projects to maintain content integrity.

# Future Modules:

# Additional features like collaboration tools for group projects and real-time notifications for updates and approvals will be introduced. These enhancements aim to improve user interaction and engagement on the platform.

# 8.IMPLEMENTATION

## SOURCE CODE:

**url.py:**

from django.contrib import admin

from django.urls import path

from django.conf.urls.static import static

from django.conf import settings

from Doctor import views as Doctor

from admins import views as admins

from forensic import views as frnsc

urlpatterns = [

path('admin/', admin.site.urls),

path('index/', Doctor.index, name='index'),

path('doctorlogin/',Doctor.doctorlogin,name='doctorlogin'),

path('doctorregister/',Doctor.doctorregister,name='doctorregister'),

path('doctorlogincheck/',Doctor.doctorlogincheck,name='doctorlogincheck'),

path('bodydetails1/',Doctor.bodydetails1,name='bodydetails1'),

path('doctorreport/',Doctor.doctorreport,name='doctorreport'),

path('doctorreport1/',Doctor.doctorreport1,name='doctorreport1'),

path('doctorfinalreport/',Doctor.doctorfinalreport,name='doctorfinalreport'),

path('frnsclogin/', frnsc.frnsclogin, name='frnsclogin'),

path('frnscloginentered/', frnsc.frnscloginentered, name='frnscloginentered'),

path('bodydetails/', frnsc.bodydetails, name='bodydetails'),

path('forensicreport/', frnsc.forensicreport, name='forensicreport'),

path('frnscreport1/', frnsc.frnscreport1, name='frnscreport1'),

path('admin1/', admins.adminlogin, name='admin1'),

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

path('doctordetails/', admins.doctordetails, name='doctordetails'),

path('activatedoctor/',admins.activatedoctor,name='activatedoctor'),

path('finalreport/',admins.finalreport,name='finalreport'),

path('adminbodydetails/',admins.adminbodydetails,name='adminbodydetails'),

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

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL,document\_root=settings.MEDIA\_ROOT)

**views.py:**

from django.shortcuts import render

from django.contrib import messages

from django.http import HttpResponse

from Doctor.models import doctorModel,doctorreportmodel

from Doctor.forms import doctorForm

from forensic.models import \*

from forensic.forms import \*

import speech\_recognition as sr

import pyttsx3

def index(request):

return render(request,'index.html')

def doctorlogin(request):

return render(request,'doctor/doctorlogin.html')

def doctorregister(request):

if request.method == 'POST':

form1 = doctorForm(request.POST)

if form1.is\_valid():

form1.save()

print("succesfully saved the data")

return render(request, 'doctor/doctorlogin.html')

# return HttpResponse("registreration succesfully completed")

else:

print("form not valied")

return HttpResponse("form not valied")

else:

form = doctorForm()

return render(request, "doctor/doctorregister.html", {"form": form})

def doctorlogincheck(request):

if request.method == 'POST':

sname = request.POST.get('email')

print(sname)

spasswd = request.POST.get('upasswd')

print(spasswd)

try:

check = doctorModel.objects.get(email=sname,passwd=spasswd)

# print('usid',usid,'pswd',pswd)

print(check)

request.session['name'] = check.name

print("name",check.name)

status = check.status

print('status',status)

if status == "Activated":

request.session['email'] = check.email

return render(request, 'doctor/doctorpage.html')

else:

messages.success(request, 'doctor is not activated')

return render(request, 'doctor/doctorlogin.html')

except Exception as e:

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

pass

messages.success(request,'Invalid name and password')

return render(request,'doctor/doctorlogin.html')

def bodydetails1(request):

qs=bodydetails.objects.all()

return render(request,'doctor/bodydetails1.html',{"object":qs})

def doctorreport(request):

return render(request,'doctor/doctorreport.html')

def doctorreport1(request):

bid=request.POST.get('t1')

qs=bodydetails.objects.filter(bid=bid)

print("qs:",qs)

return render(request,'doctor/doctorreport1.html',{"object":qs})

def doctorfinalreport(request):

r = sr.Recognizer()

print("r:", r)

# Function to convert text to

# speech

def SpeakText(command):

# Initialize the engine

engine = pyttsx3.init()

print("engine:", engine)

engine.say(command)

print("hello:", engine.say(command))

engine.runAndWait()

print(engine.runAndWait())

# Loop infinitely for user to

# speak

SpeakText("report submitted")

if request.method == 'POST':

name1=request.POST.get('t1')

print("name",name1)

bid=request.POST.get('bid')

age=request.POST.get('age')

gender=request.POST.get('gender')

ldh=request.POST.get('ldh')

print("ldh",type(ldh))

ast=request.POST.get('ast')

print("ast",type(ast))

triglycerides=request.POST.get('triglycerides')

phlevel=request.POST.get('phlevel')

rpt=int(ast)+int(triglycerides)+int(ldh)

print("rpt:",type(rpt))

phlevel=int(phlevel)

if rpt<=500 and phlevel==7:

pmi='24hrs'

elif rpt>=550 and phlevel<7 and phlevel>=5:

pmi='48hrs'

elif rpt>=550 and phlevel<5 and phlevel>=4:

pmi='72hrs'

else:

pmi='more than 72hrs'

pmi=str(pmi)

print(type(pmi))

ldh=str(ldh)

ast=str(ast)

triglycerides=str(triglycerides)

doctorreportmodel.objects.create(name=name1,bid=bid,age=age,gender=gender,ldh=ldh,ast=ast,triglycerides=triglycerides, phlevel=phlevel,pmi=pmi)

return render(request, 'doctor/doctorreport1.html')

**models.py:**

from django.db import models

# Create your models here.

class doctorModel(models.Model):

name = models.CharField(max\_length=50)

email = models.EmailField()

passwd = models.CharField(max\_length=40)

cwpasswd = models.CharField(max\_length=40)

mobileno = models.CharField(max\_length=50, default="", editable=True)

specialization = models.CharField(max\_length=50, default="", editable=True)

status = models.CharField(max\_length=40, default="", editable=True)

def \_\_str\_\_(self):

return self.email

class Meta:

db\_table='doctorregister'

class doctorreportmodel(models.Model):

name = models.CharField(max\_length=50)

bid = models.CharField(max\_length=40, default="",editable=True,unique=True)

age = models.CharField(max\_length=40)

gender = models.CharField(max\_length=40)

ldh = models.CharField(max\_length=40)

ast = models.CharField(max\_length=40)

triglycerides = models.CharField(max\_length=40)

phlevel = models.CharField(max\_length=40)

pmi = models.CharField(max\_length=40)

def \_\_str\_\_(self):

return self.bid

class Meta:

db\_table='doctorreport'

**forms.py:**

from django import forms

from Doctor.models import \*

from django.core import validators

class doctorForm(forms.ModelForm):

name = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100,)

passwd = forms.CharField(widget=forms.PasswordInput(), required=True, max\_length=100)

cwpasswd = 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.MinLengthValidator(10)])

specialization= forms.CharField(widget=forms.TextInput(), required=True, max\_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

def \_\_str\_\_(self):

return self.email

class Meta:

model=doctorModel

fields=['name','passwd','cwpasswd','email','mobileno','specialization','status']

**doctorlogin.html:**

{% extends 'base.html' %}

{% load static %}

{% block contents %}

<section id="intro" class="clearfix">

<div class="container d-flex h-100">

<div class="row justify-content-center align-self-center">

<div class="col-md-6 intro-info order-md-first order-last">

<form action="/doctorlogincheck/" method="POST">

{% csrf\_token %}<center>

<table><tr><td style="color:blue">

<!-- <input type="text" name="uname" placeholder="student Login" ><br/><br/>-->

<input type="email" name="email" placeholder="Email"><br/>

<input type="password" name="upasswd" placeholder="Password" ><br/><br/>

</td></tr></table> <input type="submit" style="color:#FF4500" value="submit"><br/>

<br><br>New Doctor Register Here ?<a href="{% url 'doctorregister' %}" style="color:blue">Click Here</a>

</center> </form> </h2>

{% if messages %}

{% for message in messages %}

<font color='darkred'> {{ message }}</font>

{% endfor %}

{% endif %}

<!--<div>

<a href="#about" class="btn-get-started scrollto">Get Started</a>

</div>-->

</div>

<div class="col-md-6 intro-img order-md-last order-first">

<img src="{% static 'img/intro-img.svg' %}" alt="" class="img-fluid"></br></br>

</div>

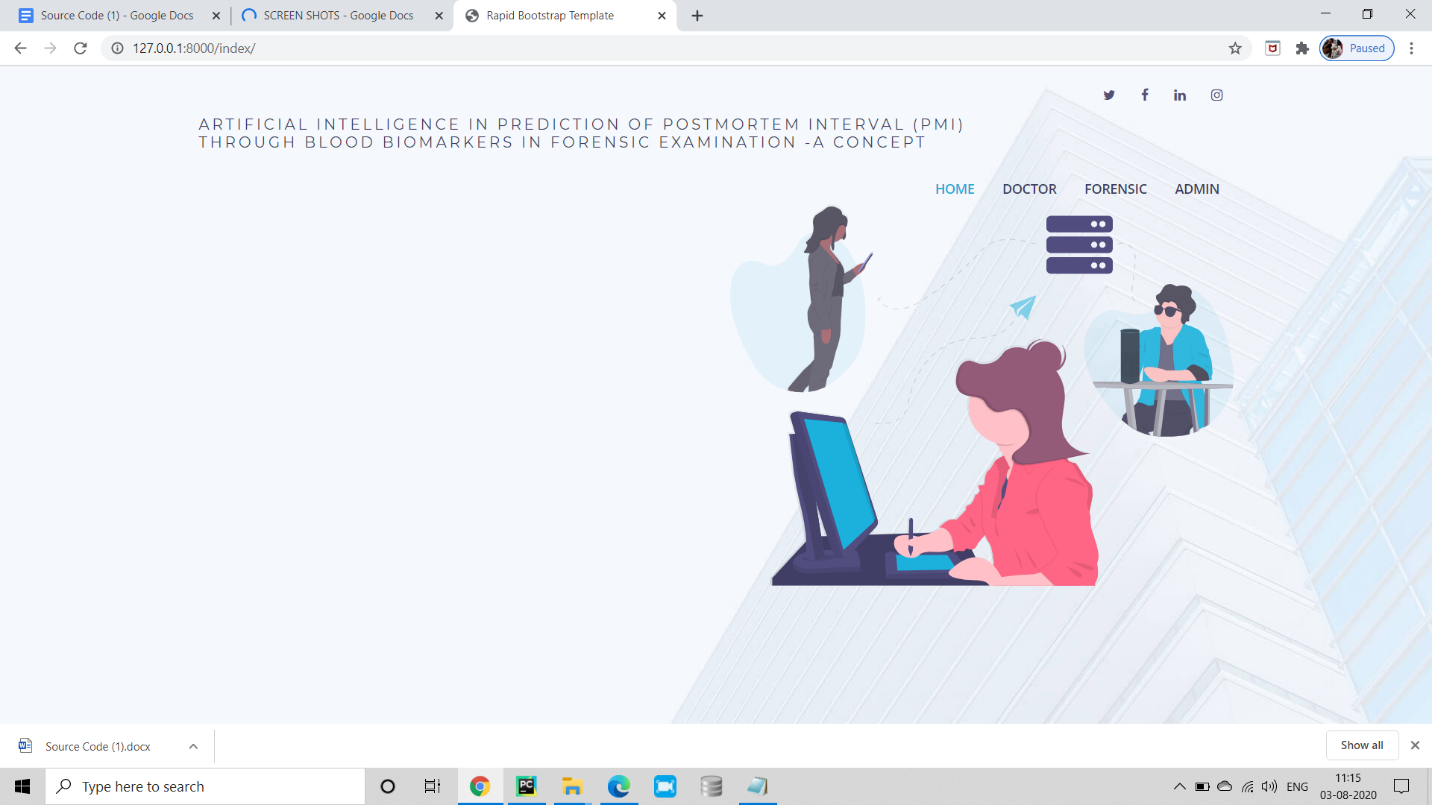
</div>

</div>

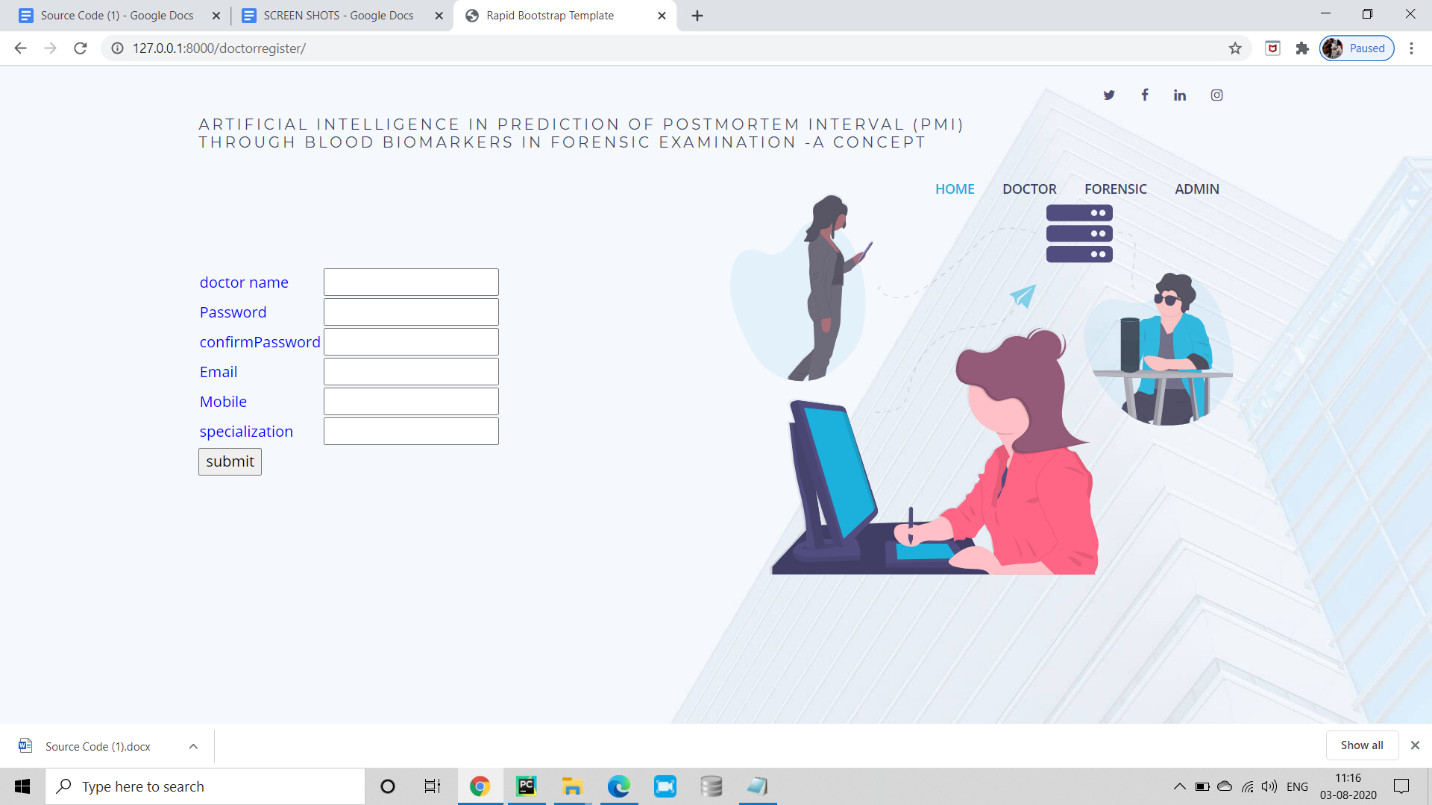
</section><!-- #intro -->

{% endblock %}

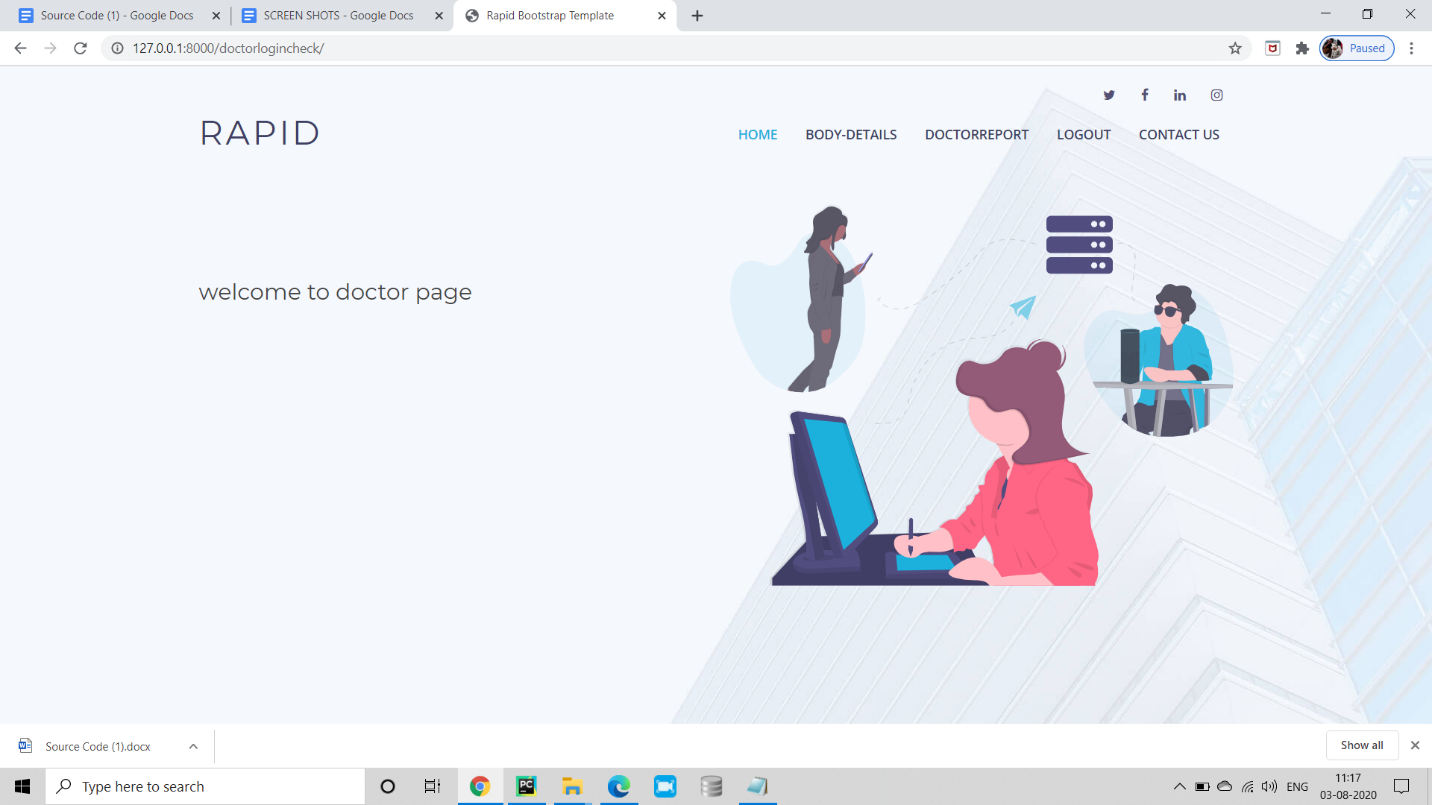
## 9. SCREENSHOTS

****

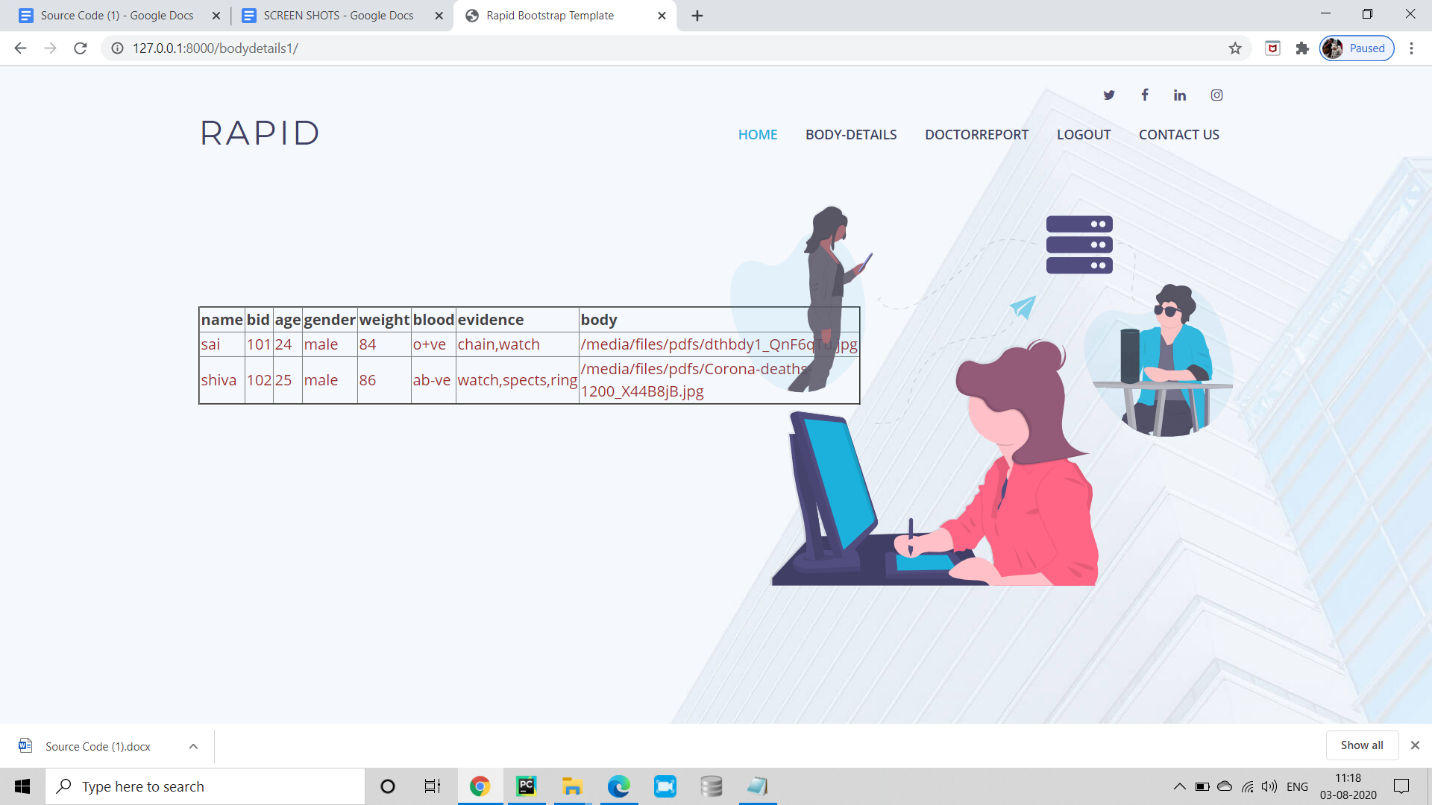
**Doctor register:**

****

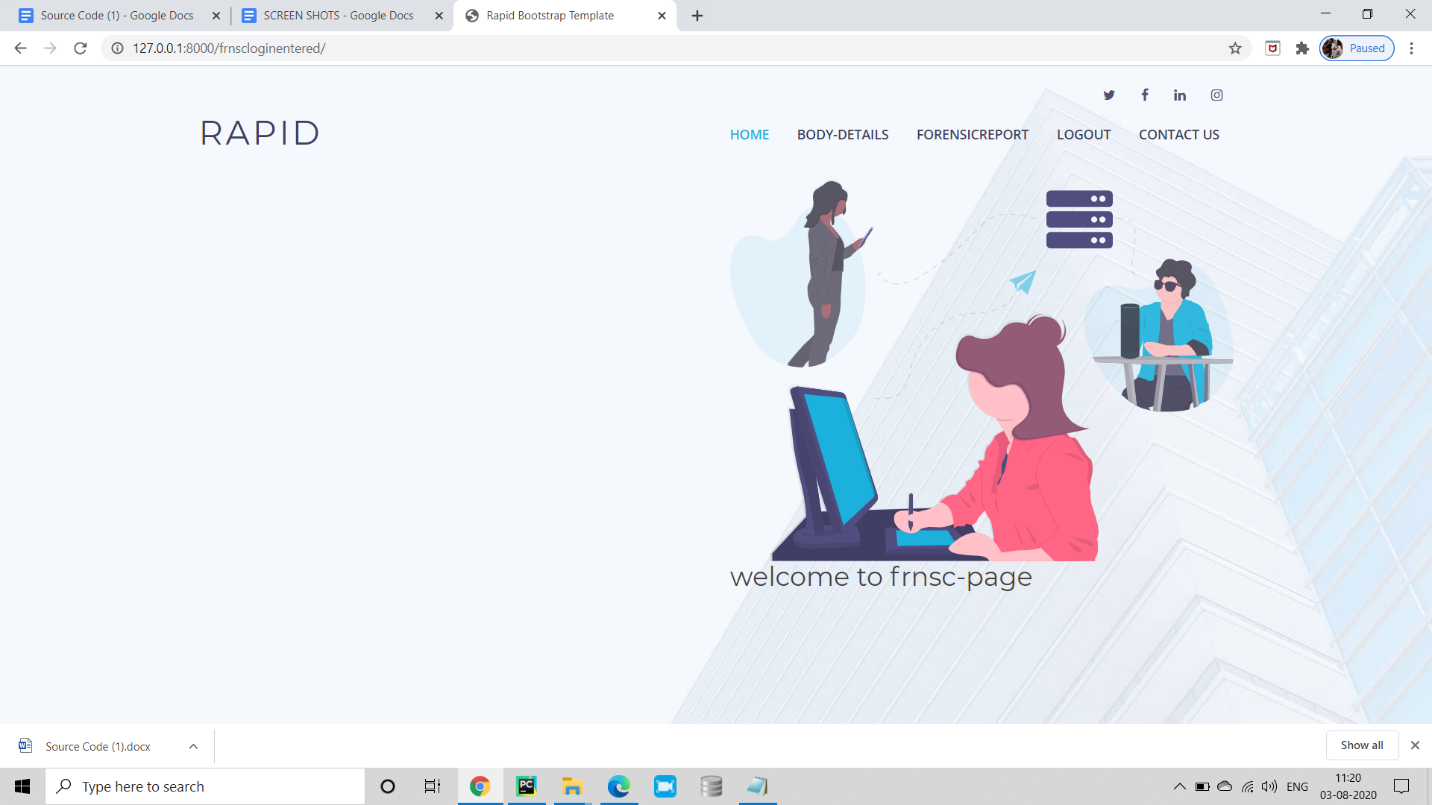
**Doctor Home:**

****

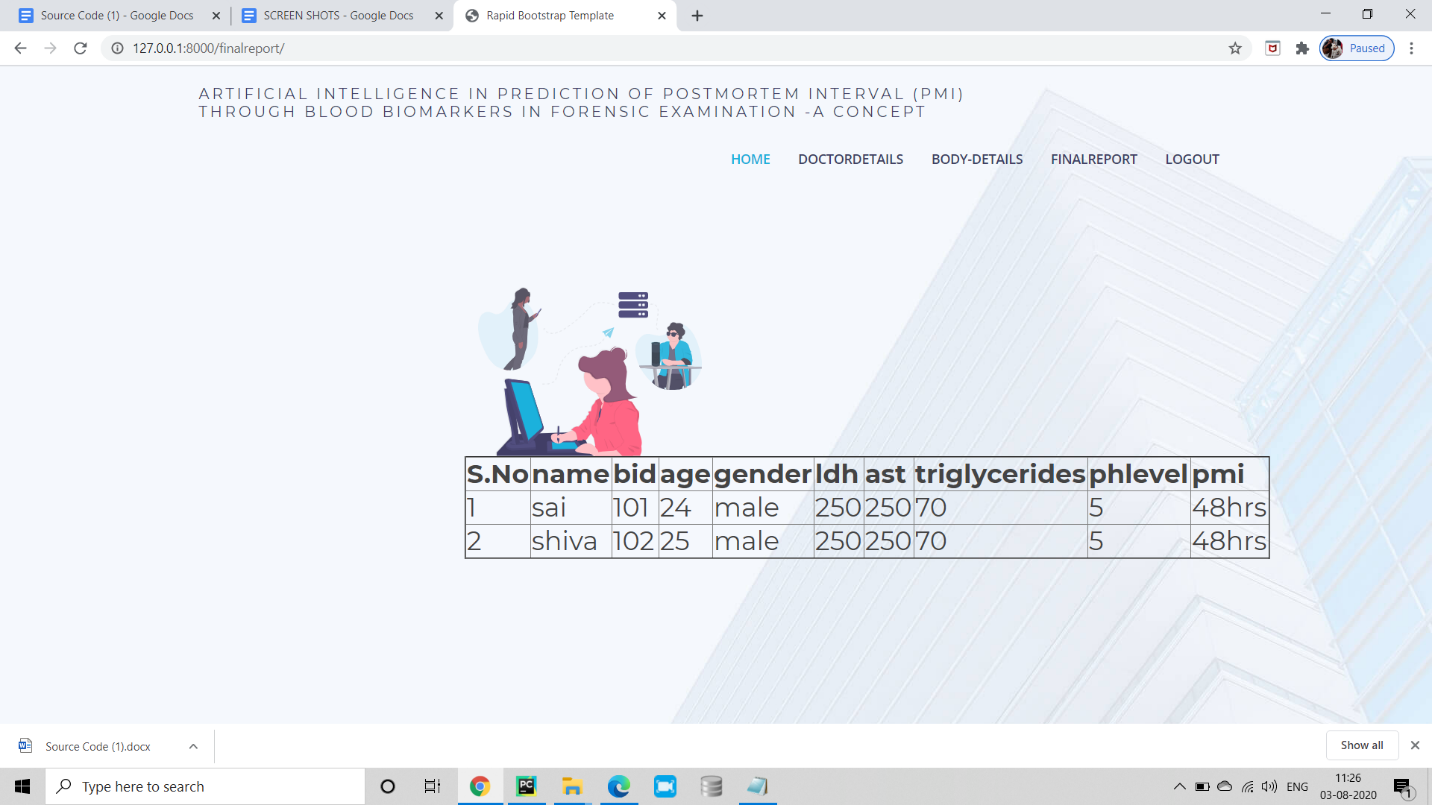
**body details:**

****

**frnsc login:**

****

**final report:**

****

# 10.TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

#### TYPES OF TESTS

##### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is a structural testing.

##### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields

Functional testing is centered on the following items:

|  |  |
| --- | --- |
| Valid Input | identified classes of valid input must be accepted. |
| Invalid Input | identified classes of invalid input must be rejected. |
| Functions | identified functions must be exercised. |
| Output | identified classes of application outputs must be exercised. |

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

##### System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration point

##### Features to be tested

* + - * Verify that the entries are of the correct format
      * No duplicate entries should be allowed

##### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

* + 1. **SAMPLE TEST CASES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **Test Case** | **Excepted Result** | **Result** | **Remarks(IF Fails)** |
| 1. | |  | | --- | |  |  |  | | --- | | User Registration | | If the user registers successfully, their account is created. | Pass | If the email already exists, registration fails. |
| 2. | |  | | --- | |  |  |  | | --- | | User Login | | If the username and password are correct, the user logs in. | Pass | Invalid credentials prevent login. |
| 3. | |  | | --- | |  |  |  | | --- | | Project Upload | | Users should be able to upload projects successfully. | Pass | If file format is incorrect, upload fails. |
| 4 | Project Search | Users can search for projects using keywords or filters. | Pass | If no matching projects exist, no results are shown. |
| 5. | |  | | --- | |  |  |  | | --- | | Project Download | | Registered users can download shared projects. | Pass | Unauthorized users cannot download projects. |
| 6. | |  | | --- | |  |  |  | | --- | | Rating & Feedback | | Users can rate and review projects. | Pass | If rating submission fails, feedback is not stored. |
| 7. | Admin Login | Admin logs in with valid credentials. | Pass | Invalid login details prevent access. |
| 8. | Admin Approval | after admin login  he will provide final report | Pass | If admin does not approve, project remains unpublished. |
| 9. | User Profile Update | Users can update profile details. | Pass | If mandatory fields are missing, update fails. |
| 10. | |  | | --- | |  |  |  | | --- | | Comment System | | Users can comment on projects. | Pass | |  | | --- | |  |  |  | | --- | | If a comment fails to  Post,an error message  appears | |

# 11.CONCLUSION

ShareNest provides an innovative solution for B.Tech students to share their projects and enhance their professional profiles. By utilizing the latest web technologies and focusing on user experience, ShareNest aims to bridge the gap between academic projects and professional opportunities. The platform promises to deliver a high-performance technology that fosters innovation and impact.

12.REFERENCES

1. <https://ieeexplore.ieee.org/document/8701416/?utm_source=chatgpt.com>.
2. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11580817/?utm_source=chatgpt.com>.
3. <https://biolres.biomedcentral.com/articles/10.1186/s40659-024-00552-8?utm_source=chatgpt.com>.
4. <https://zkginternational.com/archive/volume8/Forecast-of-PostMortem-Interval-Through-Blood-Biomarkers-Using-Artificial-intelligence.pdf>
5. <https://scimeetings.acs.org/exhibit/Prediction-postmortem-interval-PMI-through/3588959>

[6] <https://www.semanticscholar.org/paper/Artificial-Intelligence-in-Prediction-of-Post-%28PMI%29-Laxmi-Renuka/5a7cb57fcc88776f13b3dc3df06ad98ea209d3b1>.

[7]<https://www.researchgate.net/publication/332756217_Artificial_Intelligence_in_Prediction_of_PostMortem_Interval_PMI_Through_Blood_Biomarkers_in_Forensic_Examination-A_Concept>.