INTERNSHIP REPORT

On

GUESTBOOK

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

KANTA SHOWRIBABU
Regd. No: 20JR1A05A5

Under Supervision of

Mr. Madhu Parvathaneni, CEO
MadBlocks Technologies pvt.Ltd, Hyderabad.

(Duration: 8th Aug 2022 to 22nd Oct 2022)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES (ATUONOMOUS)

(APPROVED BY AICTERAND PERMANENTELY AFFILIATED TO JNTUK)
Accredited by NBA and NAAC with 'A' Grade
Vinjamanpadu (V), Vaticherkuru (M), GUNTUR – 522017

NOVEMBER-2022

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(ATUONOMOUS)

(Accredited by NBA||Accredited with„A"Grade by NAAC||Approved by A.I.C.T.E New Delhi || Affiliated to JNTUK, Kakinada) Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT),A.P-522017.

CERTIFICATE

This is to certify that the virtual industrial internship **"GUESTBOOK"** is a bonafide work of **KANTA SHOWRIBABU (20JR1A05A5),** who carried out the work under my supervisionand submitted in partial fulfillment of the requirements for the award of credits in 3-1 of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** during the Academic Year 2022-2023

HEAD OF THE DEPARTMENT

INTERNSHIP CO-ORDINATOR

EXTERNAL EXAMINER

Be an Innovator ! Be a Doer! Be an Engineer!

madBlocks
Technology:Innovation:Business

Certificate of Internship

This is certify that **Mr. KANTA SHOWRIBAU** bearing Roll No: **20JR1A05A5** from department of Computer Science and Engineering, KKR & KSR Institute of Technology and Sciences, Guntur had completed the internship with us as a Full-Stack Development intern and successfully completed the project Proof-Of-Concept (POC) with satisfactory results during the dates of **8th August to 22nd October 2022**.

MADBLOCKS TECHNOLOGIES PVT LTD

MADHU PARVATHANENI
DIRECTOR & CEO
MADBLOCKS TECHNOLOGIES PVT LTD

#20, RAINTREE PARK,
HITECH CITY ROAD,
KPHB, HYDERABAD - 500072

DECLARATION

This is to certify that the virtual industrial internship entitled "GUESTBOOK" has been carried out and submitted in partial fulfilment for the award to the Degree of **Bachelor of Technology in Computer Science and Engineering** to **Jawaharlal Nehru Technological University Kakinada** under the guidance of **MRS.FARZANA** . The work embodied in this internship work is original and has not been submitted in part or full for any degree of this or any degree of any other university.

By

KANTA SHOWRIBABU

(20JR1A05A5)

ACKNOWLEDGEMENT

I would like to express our profound gratitude towards **MRS.FARZANA,** who played a supervisory role to utmost perfection, enabled us to seek through our Skill Oriented main project and for guiding as an internal guide methodically and meticulously.

I am highly indebted to **Dr.P.BABU**, Principal, for providing us with all the necessary support.

I render our deep sense of gratitude to **Prof. R. RAMESH**, **Head of the Department,** for permitting us to carry out our main project works

I would also like to thank our parents and friends for their enduring encouragement and assistance whenever required.

By

KANTA SHOWRIBABU

(20JR1A05A5)

ABSTRACT

GuestBook describes a method of collecting the feedback from the guest through online mode. In previous days feedback can be manipulated in the process of manual work, to avoid the drawbacks we proposed a web-based feedback collecting system. The feedback collecting system is mainly concentrate on gather information from the guests or visitors about the event , college and students. The collected information is stored in the database (MYSQL). The front-end can communicate with database by using API called Python-Flask. This approach has Registration , Login , Feedback ,Database and Review modules. The registered details are stored in database and these are used at the time of login. The user can review the feedback from database by calling with the guest name. The proposed system is very effective ,easy and take less time to organize the accurate feedbacks without human intervention.

Organization Information:

MADBLOCKS is a leading student startup consulting in Telangana and Andhra Pradesh and incubating the startups based on Virtual Reality, Augmented Reality, Robotics, IoT , cloud, Machine Learning, and Artificial Intelligence. MADBLOCKS have 10+ years of experience and we always thrive for better growth and success of incubating's. MadBlocks Technologies Private Limited mission is to build a community which recreates the innovation in campus and these innovations leads to great potential startups in campus. MADBLOCKS is on a commitment to deliver our strengths which helps their clients to empower their stakeholders to reach greater heights which leads to better society and better living.

Programs and opportunities:

This ground up approach helps us deliver not only the solution to our clients but also add value to at the core. MADBLOCKS operates on various domains namely Virtual Reality, Augmented Reality, Robotics, IoT , cloud, Machine Learning, and Artificial Intelligence..

Methodologies:

We follow a structured methodology for our projects which starts from designing the solution to the implementation phase. Well planned Project reduces the time to deliver the project and any additional ad-hoc costs to our clients, hence we dedicate majority of our time understanding our clients business and gather requirements. This ground up approach helps us deliver not only the solution to our clients but also add value to your investments.

Key parts of the report:

Under each division we further provide specific industry solutions on focused domains with cutting edgetechnologies.

Benefits of the Company/Institution through our report:

Under each division we further provide specific industry solution on focused domains with cutting edge technologies. We emphasize on building relationships with our clients by delivering projects on time and within budget.

INDEX OF CONTENTS :

LIST OF FIGURES :

`

Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.

- An objective for this position should emphasize the skills you already possess in the area andyour interest in learning more

- Internships are utilized in a number of different career fields, including architecture, engineering,healthcare, economics, advertising and many more.

- Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.

- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
1st WEEK	08/08/22	Monday	Create a registration page using html
	09/08/22	Tuesday	Create a nav bar using html
	10/08/22	Wednesday	Create portfolio using html
	11/08/22	Thursday	Create a html page using frames
	12/08/22	Friday	Create interactive web page using javascript
	13/08/22	Saturday	Using all mouse events in javascript

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
2nd WEEK	15/08/22	Monday	Create forms using bootstrap
	16/08/22	Tuesday	Create various buttons using bootstrap
	17/08/22	Wednesday	Create courousels using bootstrap
	18/08/22	Thursday	Create cards using bootstrap
	19/08/22	Friday	Installation and setup of mysql
	20/08/22	Saturday	Creation of sql database

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
3rd WEEK	22/08/22	Monday	Perform various sql queries
	23/08/22	Tuesday	Working on Insert and Select commands
	24/08/22	Wednesday	Installation and setup of mongodb
	25/08/22	Thursday	Creation of cluster,database and collections
	26/08/22	Friday	Inserting records into collections
	27/08/22	Saturday	Performing various operations on data inserted

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
4th WEEK	29/08/22	Monday	Create Html With College And Your Name.
	30/08/22	Tuesday	Create Python Script To Host Web Server And Give Response As Your Name.
	31/08/22	Wednesday	Create Python Script To Host Web Server And Render Html Template.
	01/09/22	Thursday	Create Python Script To Host Web Server And Render Two Html Templates.
	02/09/22	Friday	Sent Some Data To Html On A Button Click.
	03/09/22	Saturday	Html Form, Name, Roll No, Button.

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
5th WEEK	05/09/22	Monday	Create MYSQL Database and Create Table with name, rollno as column names with varchar(255) and insert a sample value.
	06/09/22	Tuesday	Create python script to read the data from table.
	07/09/22	Wednesday	Publish the form data intodDatabase table.
	08/09/22	Thursday	Publish the form data into database table.
	09/09/22	Friday	Create api to read data from table using get method.
	10/09/22	Saturday	Create api to push data to table using get method.

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
6th WEEK	19/09/22	Monday	Use session storage and display the vaiues in the second html.
	20/09/22	Tuesday	Create register.html with attributes name, rollno as username and password and store them in the database(If Record Was Not Found).
	21/09/22	Wednesday	Create login.ttml and check rollno as username and password and display all records in success.html.
	22/09/22	Thursday	Create Mongodb database and create table with name,roll no & insert a simple value.
	23/09/22	Friday	Create python script to read data from table
	24/09/22	Saturday	Publish the form data into database table

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
7th WEEK	26/09/22	Monday	Create API To Read Data From Table
	27/09/22	Tuesday	Create Registration Page And Store Records If Does Not Exist
	28/09/22	Wednesday	Create Login Page And Display Records If Login Is Valid
	29/09/22	Thursday	Create A Web Server Using Node And Express Js And Return Your Name
	30/09/22	Friday	Render Html Page Using Flask Into The Server
	01/10/22	Saturday	Render Two Html Pages Using Flask Into Multiple Handlers

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
8th WEEK	03/10/22	Monday	Create An Api To Collect Data From Html Using Node
	04/10/22	Tuesday	Create An Api To Insert Data Into Sql Database Using Express
	05/10/22	Wednesday	Create A Connection Sql And A Registration Form
	06/10/22	Thursday	Create An Api To Insert Data Into Mongodb Using Postman
	07/10/22	Friday	Create An Api To Read Data From Sql Using Postman
	08/10/22	Saturday	Create An Api To Insert Data Into Sql Using Postman

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
9th WEEK	10/10/22	Monday	Design phase
	11/10/22	Tuesday	Development Phase
	12/10/22	Wednesday	Front-end development
	13/10/22	Thursday	Back-end development
	14/10/22	Friday	Database connection
	15/10/22	Saturday	API

	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
10th WEEK	17/10/22	Monday	Integration
	18/10/22	Tuesday	Integration
	19/10/22	Wednesday	Testing
	20/10/22	Thursday	Testing
	21/10/22	Friday	Presentation
	22/10/22	Saturday	Presentation

1.INTRODUCTION

As the part of the internship we improved our skills through full stack development. This course is learned from external trainer and as well as from our internal guide.

Course Learned:

The recent evolution in web technologies, there has never been a more exciting time for developers and technologists around the globe to build modern web applications. Web development is not anymore confined in the real time of pure HTML (HyperText Markup Language), CSS (Cascading Style Sheets), and JavaScript on the front-end. There is a plethora of new languages, web frameworks and tools to choose for any web application development. Although the rise of web technologies has helped to ease the application development process, it has created confusion among developers to select a perfect technology stack to start with.

Now here in the skill development course we learn about HTML, CSS, PYTHON FLASK, MYSQL and MONGODB.

1.1 Introduction of the project

As a part of internship and based on learned technologies we developed a project titled **GuestBook.** A GuestBook is a logging system that allows visitors or guests to leave a public comment. It is possible in GuestBook for visitors to express their thoughts on the website. It is an informal method of dropping off a quick message.

1.2 Existing Systems:

Coming to existing system the feedback is done by manual process. In the existing system guest can give feedback about the events, colleges and students by using pen and paper. After giving the feedback , papers are collected and calculate the overall grade.

Fig1.1: Feedback collection manually

But collecting feedback manually is difficult and we can’t get the accurate feedback. So taking feedback manually consumes more time and getting not accurate results.

1.3 Problems of the Existing System?

Existing System requires more time to do a piece of work . But collecting feedback manually is difficult and we can't get the accurate feedback.

Fig 1.2: collecting feedback from guests.

So taking feedback manually consumes more time and getting not accurate results. For this reason, the onweb-basedback system is implemented.

1.4 Proposed System

The proposed system is titled "**GuestBook**". The main theme of the proposed system is collecting feedback through online mode. The feedback system mainly concentrates on gather information from the guests about the events, college and students.

Users and guests after login into the system only it is possible to give and view feedback. Before login into the GuestBook the user and the guest need to register with their basic credentials . After Guests have to login with their basic details like name, password phone number and Users has login with name, password.

Fig 1.3: feedback collection from guests through online

After guest logged into GuestBook guest can fill the required columns like name , file and feedback. After click on the submit button guest receives the message like thank you for your feedback. Then the user can login and review the feedback based on the name of the guest.

1.5 Benefits of the Proposed System

- It is very effective.
- It is very easy to maintain.
- It is user friendly.
- We got the accurate feedbacks only.
- It can reach to every one.
- Review of guest also based on selection only.

1.6 Why should everyone use GUESTBOOK?

The main theme of the GuestBook is to reduce the burdens. Whenever the guest give feedback through online for any event then they can generate the report easily. As per the proposed system it is easy to manage everything related to feedbacks.

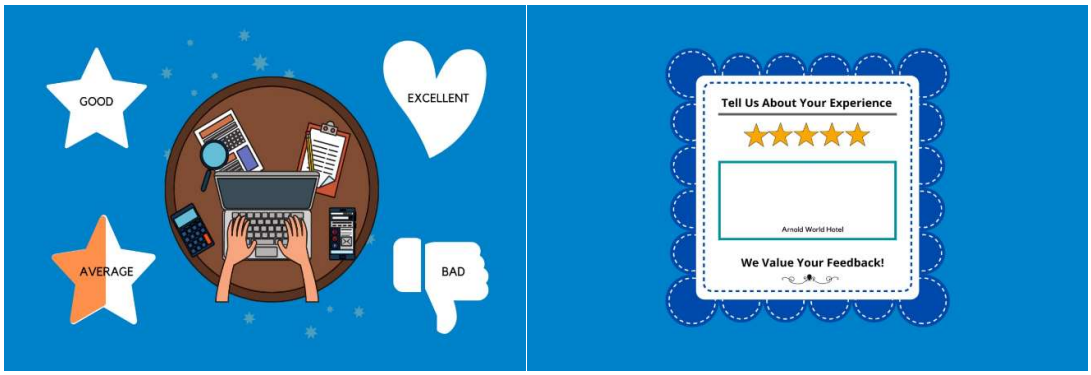


Fig 1.4: Accurate feedback from guests

We will get the accurate feedback only. Based on the accurate we can take the correct decisions. And if any problems further we can't repeat again. Take the feedback through online will get from the bottom of the heart. Accurate feedback give good results in future also.

2.TECHNOLOGIES LEARNED

At the time of skill orientation I acquired the knowledge on HTML , CSS , PYTHON -FLASK, MYSQL and MongoDB.

2.1Technologies used :

2.1.1 HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML .

2.1.3 PYTHON-FLASK

An API (Application Programming Interface) is a set of rules that are shared by a particular service. These rules determine in which format and with which command set your application can access the service, as well as what data this service can return in the response. The API acts as a layer between your application and external service.

2.1.4 MYSQL

MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it's clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL.

2.1.5 MongoDB

MongoDB is an open-source, non-relational, document database. The concept of rows still exists in MongoDB but it is called a document. A set of documents is called a collection and MongoDB holds a set of collections. The format in which MongoDB stores the data is called BSON, which stands for binary JSON.

3. SYSTEM ANALYSIS

3.1 Requirements Analysis:

Requirements analysis, also called requirements engineering. It is the process of Identifying the user expectations for a new or modified product. These features called requirements must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups, avoidance of feature creep and documentation of all aspects of the project development process from start to finish.It is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

3.1.1 Functional Requirements Analysis:

It is a useful document which describes functions, appearance, purpose and requested outputs of the software. It allows you to structure all the information regarding an application.

3.1.2 Users Requirements:

Generally the college students and the management are wait for the guest feedback. But in previous days the feedback can collected through pen and paper ,but it not gives the accurate results. Because it is offline mode so guest submit the feedback manually ,it any negative feedback will give bad impression on guest ,so they also don't like to give accurate feedbacks.

So our proposed system can collect the feedback through online mode ,we will confidently says that our proposed system gives accurate results.

Internal Users:

- No internal users

External Users:

- Students or a guests who wants to express their views and for reviews .

3.2 Non-Functional Requirements:

Availability:

It is an open-source website which can be accessed by any registered user.

Usability:

It is very easy to use as everyone is familiar with reading and using many more websites regularly.

Performance:

The performance is good as all the requirements of users are embedded in our website.

Reliability:

Our website has ability to perform its intended functions and operations in a system without any failure.

3.3 Process Model:

The **prototype model** is designed using **structured modeling** and is able to provide the desired results. It can be successfully implemented as a Real Time system with certain modifications. Science is discovering or creating major breakthrough in various fields, and hence technology keeps changing from time to time. Going further, most of the units can be fabricated on a single along with microcontroller thus making the system compact thereby making the existing system more effective. To make the system applicable for real time purposes components with greater range needs to be implemented.

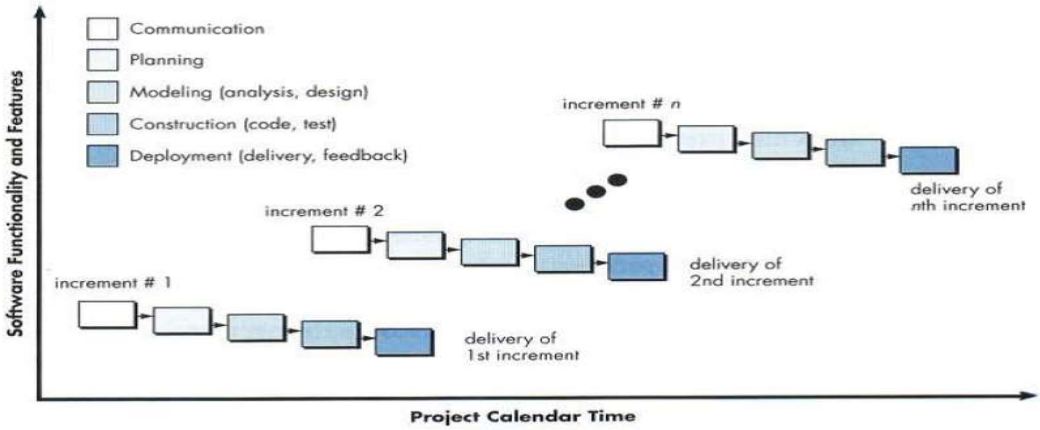


Fig 3.1:incremental process model

Principles of structured modeling:

- Structured Modeling is an attempt to redress this imbalance. Structured Modeling aims to provide a formal mathematical framework and computer–based environment for conceiving, representing and manipulating a wide variety of models.
- To establish close contact with the customer during development and to gain a clear understanding of various requirements, each agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review the progress made and re-evaluate the requirements.
- Frequent delivery of incremental versions of the software to the customer representative in intervals of a few of a few weeks.
- structured development processes usually deploy Pair Programming. In Pair programming, two programmers work together at one work-station. One does coding while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.

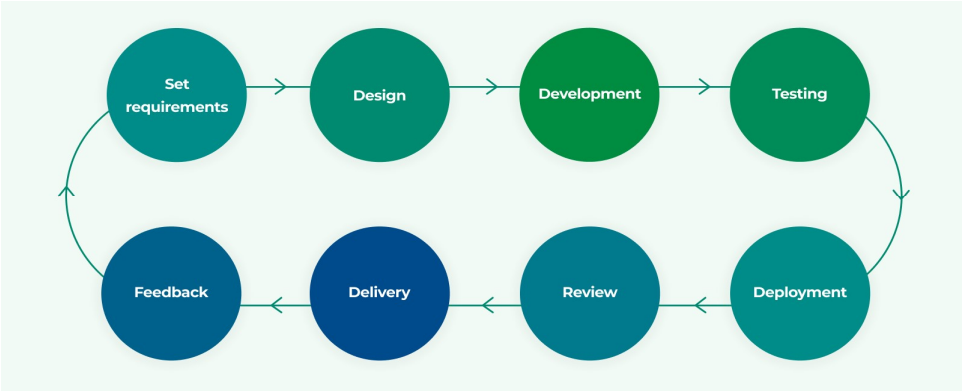


Fig 3.2: structure model

4.MODULES DESCRIPTION

We have divided our project into 2 modules. They are

- User module
- Guest module

4.1 Guest Module :

The guest module consists of guest registration and guest login. The guest can register with his basic credentials like name , phone number ,password.

Guest can login by using name and passwords and guest can give feedback by providing their name and image and feedback about the college and students and event and guest also give feedback about how the students and management received and treat them .

Fig 4.1: guest login

4.2 User Module :

The user module consists of user registration and user login. The user can register with his basic credentials like name ,roll number ,password and branch.User can login by using name and passwords and user can able review the feedback of the guest based on the guest name.

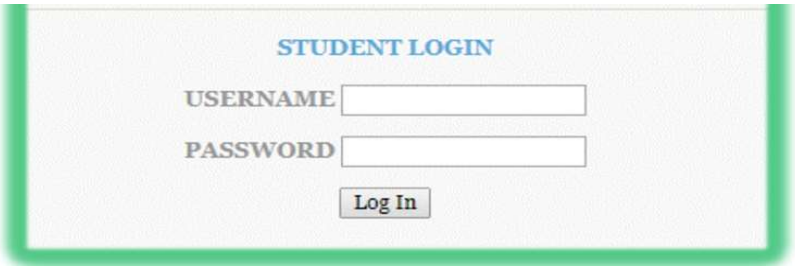


Fig 4.2 : Student Login

4.3 Feasibility Study

A feasibility study is a study usually done by engineers, which establishes whether conditions are right to implement a particular project. Feasibility studies can be done for many purposes, and are sometimes done in IT in order to look at feasibility for new hardware and software setups sometimes a feasibility study is done as part of a systems development life cycle, in order to drive precision for the implementation of technologies. Engineers might look at a five-point model called TELOS this includes the following components:

- Technical
- Operational
- Behavioral
- Process model used
- Hardware requirements and Software requirements
- System requirement specification

4.3.1 Behavioral Feasibility:

- It performs the intended functionalities and operations.

4.3.2 Economic Feasibility:

- Users have no charge using this website.

5. HARDWARE REQUIREMENTS AND SOFTWARE REQUIREMENTS

5.1 Hardware requirement:

- OS : Windows 10 / Linux
- Hard disk : 20GB
- RAM : 8GB

5.2 Software Requirements:

- Any web browser
- Visual studio code
- Python
- Mysql
- MongoDB

5.3 Software Requirements Specification

- SRS is a software requirement specification.
- SRS act as an agreement between the client and business manageger.
- And SRS is basically an organization and understanding of a client and system requirements and dependencies at a particular point in time.
- The SRS document itself states in precise and explicit language those functions and capabilities a software system must provide, as well as states any required constraints by which the system must abide.
- Software requirements specification establishes the basis for an agreement between customers and developers or suppliers on how the software product should function Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign.
- It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.
- The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development.

This is achieved through detailed and continuous communications with the project team and customer throughout the software development process here needs to change.

6.DESIGN PHASE

6.1Design concepts

The set of fundamental software design concepts are as follows:

Abstraction

- Environment at the highest-level abstraction
- The lower level of abstraction provides a more detail description of the solution
- A sequence of instruction that contain a specific and limited function refers in a procedural abstraction

Architecture

- The complete structure of the software is known as software architecture
- Structure provides conceptual integrity for a system
- The architecture is the structure of program modules.
- The aim of the software design is to obtain an architectural framework of a system.

Patterns

- A design pattern describes a design structure that solves a particular design problem in a specified context.

Modularity

- Software is separately divided into name and addressable components; they are called as modules which makes design easy.
- Modularity is the single attribute of software that permits a program to be managed easily.

Information hiding

- Modules must be specified and designed so that the information like algorithm and data presented in a module is not accessible for other modules which do not require that information.

Functional independence

- The functional independence is the concept of separation and related to the concept of modularity, abstraction and information hiding
- The functional independence is accessed using two criteria i.e., Cohesion and coupling

Cohesion

- Cohesion is an extension of the information hiding concept
- A cohesive module performs a single task and it requires a small interaction with the other components in other parts of the program

Coupling

- Coupling is an indication of interconnection between modules in a structure of Software

6.2 Design concepts for our project:

Abstraction:

We only display the statistical data, hiding the raw data and background implementation of the storage of data.

Modularity:

Our total project is divided into various modules and Integrated later in order to make the Implementation easier.

6.3 Design Constraints of our project:

Non - Functional requirements:

Non-functional requirements in our project include Security, Availability, and performance.

Commercial Constraint:

The budget required to develop this project is minimum and also the time required for completion is 1 month.

6.4 Logical Design :

The logical design describes the execution flow of our project.

6.4.1 Flowchart :

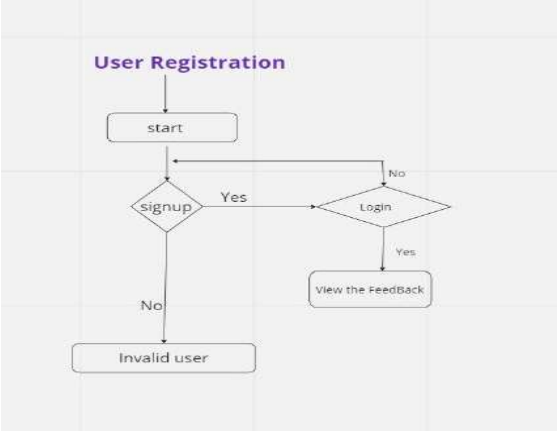


Fig 6.1: user flow chart

The user registration and user login are describes in above flow chart. The user can register with his basic credentials like name ,roll number ,password and branch. If ok then allow for login. Now user can login by using name and passwords and user can able review the feedback of the guest based on the guest name.

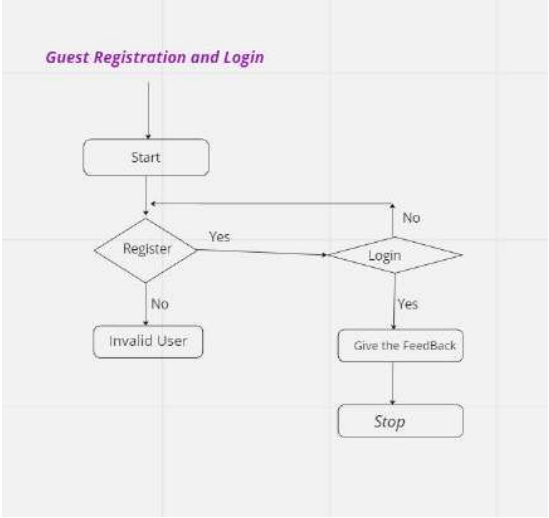


Fig 6.2 : Guest flow chart

The guest registration and guest login are describes in above flow chart. The guest can register with his basic credentials like name, phone number, password. If ok then allow for login. Now guest can login by using name and passwords and guest can give feedback by providing their name and image and feedback about the college and students and event and guest also give feedback about how the students and management received and treat them .

6.4.2 ER Diagram:

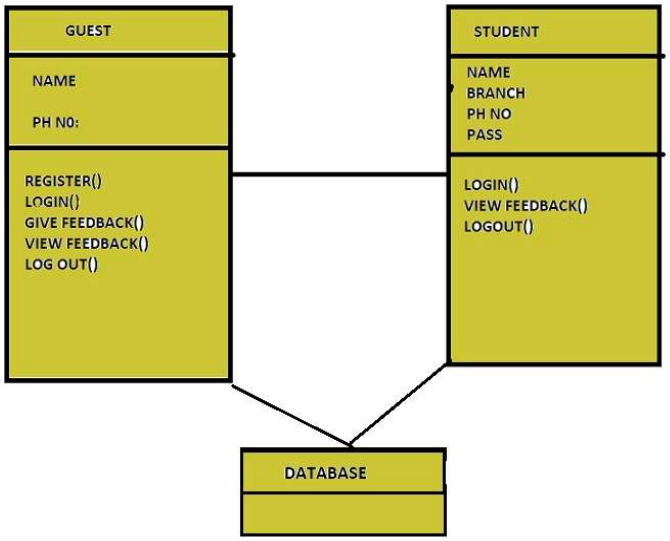


Fig 6.3 ER diagram

6.5 Physical Design:

The physical design consists of coding ,here we are using html and python flask.

Python- Flask code:

```
from fileinput import filename
from flask import
Flask,render_template,redirect,request,flash,url_for,send_file,send_from_director
y
#for file uploading..
import urllib.request
import os
from werkzeug.utils import secure_filename
import mysql.connector as mysql
#obj..
app=Flask(__name__)
mydb=mysql.connect(
    host='localhost',
    user='root',
    password='Ksb6419*',
    database='showri'
)
cursor=mydb.cursor()

UPLOAD_FOLDER='static'
app.secret_key='1111'
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
ALLOWED_EXTENSIONS=set(['png','jpg','jpeg','gif'])
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS
#filename=''
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024


#handlers...


@app.route('/')
def start():
    return render_template('home.html')

#guest....
```

```
@app.route('/guest')
def home():
    return render_template('guest.html')

@app.route('/greg')
def greg():
    return render_template('guest-reg.html')

@app.route('/glogin')
def glogin():
    return render_template('guest-sign.html')


#user...
@app.route('/user')
def user():
    return render_template('user.html')

@app.route('/ureg')
def ureg():
    return render_template('user-reg.html')

@app.route('/ulogin')
def ulogin():
    return render_template('user-sign.html')




#form data..

#user reg
@app.route('/urdata',methods=['post'])
def urdata():
    name=request.form['name'].strip()
    password=request.form['pass'].strip()
    branch=request.form['branch'].strip()
    phno=request.form['phone'].strip()
    sql='select * from user'
    cursor.execute(sql)
    d=cursor.fetchall()
```

```
    for i in d:
        if (i[1]==password and (i[0]==name.upper() or i[0]==name.lower())):
            return render_template('user-reg.html',e='user already registred')

    s='insert into user(name,password,branch,phno)values(%s,%s,%s,%s)'
    v=(name,password,branch,phno)
    cursor.execute(s,v)
    mydb.commit()
    return render_template('user-reg.html',s='signup successfully')

#user login..
@app.route('/usdata',methods=['post'])
def usdata():
    name=request.form['name'].strip()
    password=request.form['pass'].strip()
    phno=request.form['phno'].strip()
    sql='select * from user'
    cursor.execute(sql)
    d=cursor.fetchall()
    flag=0
    for i in d:
        if(i[3]==phno):

            flag=1
            return render_template('gone.html')
            #if user login success then all feedbacks will be on the user success
page..
                #return render_template('feedback.html',n=name,f=fback)
    if(flag==0):
        return render_template('user-sign.html',lf='Please signup first')



#guest one....
@app.route('/gone',methods=['post'])
def gone():
    name=request.form['name'].strip()

    #guests data..feedback
    s2='select * from feedback'
    cursor.execute(s2)
```

```
    f=cursor.fetchall()
    data=[]
    flag=0
    for j in f:
        if name==j[0]:
            flag=1
            name=j[0].strip()
            fback=j[1].strip()
            filename=j[2].strip()
            dummy=[]
            dummy.append(name)
            dummy.append(fback)
            dummy.append(filename)
            data.append(dummy)
            return render_template('gones.html',data=data)
            #print(data)
    if(flag==0):
        return render_template('gone.html',de='Not found with that Guest Name')


@app.route('/gones',methods=['post'])
def gones():
    name=request.form['name'].strip()

    #guests data..feedback
    s2='select * from feedback'
    cursor.execute(s2)
    f=cursor.fetchall()
    data=[]
    flag=0
    for j in f:
        if name==j[0]:
            flag=1
            name=j[0].strip()
            fback=j[1].strip()
            filename=j[2].strip()
            dummy=[]
            dummy.append(name)
            dummy.append(fback)
            dummy.append(filename)
            data.append(dummy)
```

```
            return render_template('gones.html',data=data)
            #print(data)
    if(flag==0):
        return render_template('gones.html',de='Guest Not Found')



#guest reg

@app.route('/grdata',methods=['post'])
def grdata():
    name=request.form['name'].strip()
    password=request.form['pass'].strip()
    topic=request.form['topic'].strip()
    phno=request.form['phone'].strip()
    sql='select * from guest'
    cursor.execute(sql)
    d=cursor.fetchall()
    for i in d:
        if (i[1]==password and (i[0]==name.upper() or i[0]==name.lower())):
            return render_template('guest-reg.html',e='You are  already
registred')

    s='insert into guest(name,password,topic,phno)values(%s,%s,%s,%s)'
    v=(name,password,topic,phno)
    cursor.execute(s,v)
    mydb.commit()
    return render_template('guest-reg.html',s='signup successfully')


@app.route('/gsdata',methods=['post'])
def gsdata():
    name=request.form['name'].strip()
    password=request.form['pass'].strip()
    sql='select * from guest'
    cursor.execute(sql)
    d=cursor.fetchall()
    for i in d:
        if(i[1]==password):
            return render_template('gfeedback.html')
    return render_template('guest-sign.html',lf='Please signup in the guest
```

```
registration')


#guest feedback...handling..
@app.route('/gf',methods=['post'])
def gf():
    #global filename
    name=request.form['name'].strip()
    fback=request.form['fd'].strip()
    #file...
    if 'file' not in request.files:
        return render_template('gfeedback.html',f=" file Not Avilable")
    file = request.files['file']
    if file.filename == '':
        return render_template('gfeedback.html',f='No image selected for
uploading')
    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        #here file secure with its name...
        file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        #file_url=url_for('get_file',filename=filename)

        #print('upload_image filename: ' + filename)
        #flash('Image successfully uploaded')
    else:
        return render_template('gfeedback.html',f="'Allowed image types are -
png, jpg, jpeg, gif'")
    #file uploaded ...into static/folder name
    s='insert into feedback(name,fback,filename)values(%s,%s,%s)'
    v=(name,fback,str(filename).strip())
    cursor.execute(s,v)
    mydb.commit()
    return render_template('gfeedback.html',s='Thank you for your
feedback')#,filename=filename



#guest book..
@app.route('/guestbook')
def guestbook():
    s2='select * from feedback'
```

```
    cursor.execute(s2)
    f=cursor.fetchall()
    data=[]
    for j in f:
        name=j[0].strip()
        fback=j[1].strip()
        filename=j[2].strip()
        dummy=[]
        dummy.append(name)
        dummy.append(fback)
        dummy.append(filename)
        data.append(dummy)
    return render_template('guestbook.html',data=data)



#server start..

if __name__=='__main__':
    app.run(debug=True)
```

Feedback.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login success page</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
```

```
background: rgba(0, 0, 0, 0.5);

}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;
  }


}
```

```
    </style>
</head>
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
        <li style="float: right;margin-left: 80px;"> <a href="/ureg"
>userRegistration</a></li>
        <li style="float:right"><a href="/ulogin" >UserLogin</a></li>


     </ul>


    <h2>Guest feedbacks</h2>
    <table border="1px" cellpadding="10px" align="center" cellspacing="4px">
        <tr><th>Name</th><th>image</th><th>feedback</th></tr>
        <!--<tr><td></td><td></td></tr>-->

        {% for item in data %}
          <tr>
            <td>{{item[0]}}</td>
            <td height="200px" width="200"><img
src="{{url_for('static',filename=item[2])}}" alt="guest image" height="50px"
width="100px"></td>
            <td>{{item[1]}}</td>
          </tr>
          <!--from form data-->
          <!--from form feedback..
          static\files\entrance_image.jpg-->
        {%endfor%}


    </table>
</body>
</html>
```

Gfeedback.html :

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Guest feedback</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
background: rgba(0, 0, 0, 0.5);

}
.text{
  background-color:rgb(227, 199, 163);
align-items: center;
text-align: center;
color:black;
font-size: 30px;
}

body{
  background-color: peachpuff;
  overflow: visible;
}

table{
  background-color: rgb(246, 163, 133);
  align-items: center;

}
li {
  float: left;
}

li a {
```

```
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}
#ss{
  background-color:rgb(238, 187, 120);
align-items: center;
text-align: center;
color:black;
font-size: 25px;
}

#ff{

color:black;
}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;
  }
```

```
}

    </style>
</head>
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
        <li style="float: right;margin-left: 80px;"> <a href="/greg"
>Guestregistration</a></li>
        <li style="float:right"><a href="/glogin" >GuestLogin</a></li>

     </ul>


     <div class="text">
    <h3>Please give feedback about today</h3>
  </div>
    <form action="/gf" method="post"  enctype="multipart/form-data">
        <table border="1px" cellpadding="10px" align="center" cellspacing="4px">
            <tr><td>NAME:</td><td><input type="text" name="name"
required></td></tr>

            <tr><td>Select a file:</td><td><input type="file" name="file"
required autocomplete="off">

                            <!--file controals..-->
                                  <p id="ff">{{f}}</p>
                                </td></tr>

            <tr><td colspan="2"><textarea name="fd" cols="150"
rows="20"></textarea></td></tr>
            <tr id="tr"><td colspan="2" align="center" ><input type="submit"
value="submit" ></td></tr>
        </table>
    </form>

    <p id="ss">{{s}}</p>
</body>
</html>
```

User Reg.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>userRegistration</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
background: rgba(0, 0, 0, 0.5);

}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
```

```
}
body{
  background-color: darkorange;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;
  }

}
body{
  background: url('hell.gif');

  background-size: cover;
  background-attachment: fixed;
}
.book{
  margin-top: 150px;

}
#ss{
  margin-top: 50px;
background-color: rgb(210, 122, 210);
height: 45px;
align-items: center;
text-align: center;
color:black;
margin-top: 50px;
font-size: 20px;
}
#ee{
```

```
  background-color: rgb(209, 111, 209);
  height: 45px;
  align-items: center;
  text-align: center;
color:black;
text-shadow: 2px 2px 1px red;
margin-top:30px ;
font-size: 20px;
}


    </style>
</head>
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
     <li style="float: right;margin-left: 80px;"> <a href="/ureg"
>userRegistration</a></li>
        <li style="float:right;margin-left: 20px;"><a href="/ulogin"
>UserLogin</a></li>

     </ul>

    <div class="book">
      <h2 align="center">User Registration</h2>
        <table align="center" border="1px" cellpadding="5px" cellspacing="1px">
            <form action="/urdata" method="post">
                <tr><td style="font-weight: bolder;">Enter Name :</td><td><input
type="text" name="name" required></td></tr>
                <tr><td style="font-weight: bolder;"> password:</td><td><input
type="password" name="pass" required></td></tr>
                <tr><td style="font-weight: bolder;">Select
Branch:</td><td><select name="branch" >
                                                    <option
value="CSE">CSE</option>
                                                    <option
value="ECE">ECE</option>
                                                    <option
value="MECH">MECH</option>
```

```
                                                    <option
value="EEE">EEE</option>
                                                    <option
value="IT">IT</option>

                                                    </select>
                                                </td>

                <tr><td style="font-weight: bolder;">Phone number :</td><td>
<input type="text" name="phone" required> </td></tr>
                <tr id="tr"><td colspan="2" align="center" style="font-weight:
bolder;"><input type="submit" value="SIGNUP" ></td></tr>
            </form>
        </table>
    </div>


    <p id="ss">{{s}}</p>
    <p id="ee">{{e}}</p>

</body>
</html>
```

User Sign .html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>userSignin</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
```

```
background: rgba(0, 0, 0, 0.5);

}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
body{
  background:url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcT_tjInWP7Wz8dix5s98byG75d1VkEQmM1mVw&usqp=CAU
');
  background-size: cover;
  background-attachment: fixed;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
```

```
  body{
    margin:0px;
  }


}
.form{
  margin-top: 50px;
}
#lff{
  background-color:rgb(68, 169, 68);
  height: 45px;
  align-items: center;
  text-align: center;
color:black;
text-shadow: 2px 2px 1px red;
}


    </style>
</head>
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
     <li style="float: right;margin-left: 80px;"> <a href="/ureg"
>userRegistration</a></li>
        <li style="float:right;margin-left: 20px;"><a href="/ulogin"
>UserLogin</a></li>

     </ul>



    <div class="form">
      <h3 >LOGIN PAGE</h3>
        <form action="/usdata" method="post">
            <table border="1px" cellpadding="10px" align="center"
cellspacing="4px">
                <tr><td style="font-weight: bolder;">Name:</td><td><input
type="text" name="name"></td></tr>
```

```
                <tr><td style="font-weight: bolder;">PASSWORD:</td><td><input
type="password" name="pass" required></td></tr>
                <tr><td style="font-weight: bolder;">Phno:</td><td><input
type="text" name="phno" required></td></tr>

                <tr align="center"><td colspan="2" style="font-weight:
bolder;"><input type="submit" value="LOGIN"></td></tr>
            </table>
        </form>
    </div>

    <p id="lff">{{lf}}</p>


</body>
</html>
```

Guest Book.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login success page</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
background: rgba(0, 0, 0, 0.5);

}

li {
  float: left;
```

```
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;


  }


}

.ut{
color:black;
text-align: center;
background-color: rgb(241, 179, 98);
```

```
height:60px;
justify-content: center;
align-items: center;
padding: 20px;
margin-top: 20px;
text-transform: capitalize;

}
.book{
  margin-top: 30px;
  overflow:visible;
 }

.book table{
  background:linear-gradient(rgb(150, 143, 143),rgb(228, 223, 223),rgb(232, 73,
73),rgb(228, 223, 223),rgb(203, 201, 201))

}
body{
  background-color: rgb(196, 196, 61);
}


    </style>
</head>
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
        <li style="float: right;margin-left: 80px;"> <a href="/guest"
>Guest</a></li>
        <li style="float:right;margin-left: 20px;"><a href="/user" >User</a></li>


     </ul>

<div class="ut">
  <h2>Guest Feedbacks About Kits College</h2>
</div>

<div class="book">
```

```
    <table border="1px" cellpadding="15px" align="center" cellspacing="4px">
        <tr><th cellpadding="20px">Name Of the
Guest</th><th>Image</th><th>Review</th></tr>
        <!--<tr><td></td><td></td></tr>-->

        {% for item in data %}
          <tr>
            <td style="font-weight: bolder;" >{{item[0]}}</td>
            <td height="150px" width="200"><img
src="{{url_for('static',filename=item[2])}}" alt="guest image"
height="120px"  width="180px"></td>
            <td >{{item[1]}}</td>
          </tr>
          <!--from form data-->
          <!--from form feedback..
          static\files\entrance_image.jpg-->
        {%endfor%}


    </table>

  </div>
</body>
</html>
```

Guest reg.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration page</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
```

```
align-items: center;
background: rgba(0, 0, 0, 0.5);

}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;
  }


}
```

```
body{
  background: url('hell.gif’);

  background-size: cover;
  background-attachment: fixed;
}


.t{
  margin-top: 100px;
}

#ss{
  margin-top: 50px;
background-color: rgb(161, 68, 161);
height: 45px;
align-items: center;
text-align: center;
color:green;
text-shadow: 2px 2px 1px white;
}
#ee{

  background-color: rgb(161, 63, 161);
  height: 45px;
  align-items: center;
  text-align: center;
color:black;
text-shadow: 2px 2px 1px red;
}

    </style>
</head>
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
        <li style="float: right;margin-left: 80px;"> <a href="/greg"
>Guestregistration</a></li>
        <li style="float:right;margin-left: 20px;"><a href="/glogin"
```

```
>GuestLogin</a></li>

     </ul>
     <h2 align="center">Guest Registration form</h2>

    <div class="t">

        <form action="/grdata" method="post">
            <table border="1px" cellpadding="10px" align="center"
cellspacing="4px">
                <tr><td style="font-weight: bolder;">NAME:</td><td><input
type="text" name="name" required></td></tr>
                <tr><td style="font-weight: bolder;">PASSWORD:</td><td><input
type="password" name="pass" required></td></tr>
                <tr><td style="font-weight: bolder;">Select
Topic:</td><td><select name="topic" >
                                                    <option
value="seminar">seminar</option>
                                                    <option
value="training">training</option>
                                                    <option
value="hackthon">hackthon</option>
                                                    <option
value="suggestion">suggestion</option>
                                                    <option
value="speech">speech</option>
                                                    </select>
                                                </td>
                <tr><td style="font-weight: bolder;">Phone number :</td><td>
<input type="text" name="phone"> </td></tr>
                <tr id="tr"><td colspan="2" align="center" ><input type="submit"
value="SIGNUP" ></td></tr>
            </table>
        </form>

    </div>

    <p id="ss">{{s}}</p>
    <p id="ee">{{e}}</p>
```

```
</body>
</html>
```

Guest sign.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Signin page</title>
    <style>

            ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
background: rgba(0, 0, 0, 0.5);

}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;

}
body{
  background:url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcT_tjInWP7Wz8dix5s98byG75d1VkEQmM1mVw&usqp=CAU
');
  background-size: cover;
```

```
  background-attachment: fixed;
}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;
  }


}
.form{
  margin-top: 50px;
}

#lff{
  background-color:rgb(68, 169, 68);
  height: 45px;
  align-items: center;
  text-align: center;
color:black;
text-shadow: 2px 2px 1px red;
}

    </style>
</head>
```

```
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
        <li style="float: right;margin-left: 80px;"> <a href="/greg"
>Guestregistration</a></li>
        <li style="float:right;margin-left: 10px;"><a href="/glogin"
>GuestLogin</a></li>

     </ul>

     <h3 align="center">Login</h3>
    <div class="form">

        <form action="/gsdata" method="post">
            <table border="1px" cellpadding="10px" align="center"
cellspacing="4px">
                <tr><td style="font-weight: bolder;">Name:</td><td><input
type="text" name="name"></td></tr>
                <tr><td style="font-weight: bolder;">PASSWORD:</td><td><input
type="password" name="pass" required></td></tr>
                <tr align="center"><td colspan="2"><input type="submit"
value="LOGIN"></td></tr>
            </table>
        </form>
    </div>
<p id="lff">{{lf}}</p>

</body>
</html>
```

Gfeedback.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Guest feedback</title>
    <style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;

align-items: center;
background: rgba(0, 0, 0, 0.5);

}
.text{
  background-color:rgb(227, 199, 163);
align-items: center;
text-align: center;
color:black;
font-size: 30px;
}

body{
  background-color: peachpuff;
  overflow: visible;
}

table{
  background-color: rgb(246, 163, 133);
  align-items: center;

}
li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
```

```
}
#ss{
  background-color:rgb(238, 187, 120);
align-items: center;
text-align: center;
color:black;
font-size: 25px;
}

#ff{

color:black;
}

ul a:hover{
  background-color:red;
  color: white;
  opacity:0.8;
  box-shadow: 1px 2px 2px white;
}
@media screen and (max-width:990px){
  ul{
    display: flex;
padding:5px;
  }
  ul a{
    margin:0px;
  }
  li a{
    padding: 0px;
  }
  body{
    margin:0px;
  }


}

    </style>
</head>
```

```
<body>
    <ul>
        <li style="margin-left:60px;"> <a href="/" >Home</a></li>
        <li style="margin-left:100px"><a href="/guestbook" >GuestBook</a></li>
        <li style="float: right;margin-left: 80px;"> <a href="/greg"
>Guestregistration</a></li>
        <li style="float:right"><a href="/glogin" >GuestLogin</a></li>

     </ul>


     <div class="text">
    <h3>Please give feedback about today</h3>
  </div>
    <form action="/gf" method="post"  enctype="multipart/form-data">
        <table border="1px" cellpadding="10px" align="center" cellspacing="4px">
            <tr><td>NAME:</td><td><input type="text" name="name"
required></td></tr>

            <tr><td>Select a file:</td><td><input type="file" name="file"
required autocomplete="off">

                            <!--file controals..-->
                                   <p id="ff">{{f}}</p>
                                 </td></tr>

            <tr><td colspan="2"><textarea name="fd" cols="150"
rows="20"></textarea></td></tr>
            <tr id="tr"><td colspan="2" align="center" ><input type="submit"
value="submit" ></td></tr>
        </table>
    </form>

    <p id="ss">{{s}}</p>
</body>
</html>
```

7.OUTPUT SCREENS

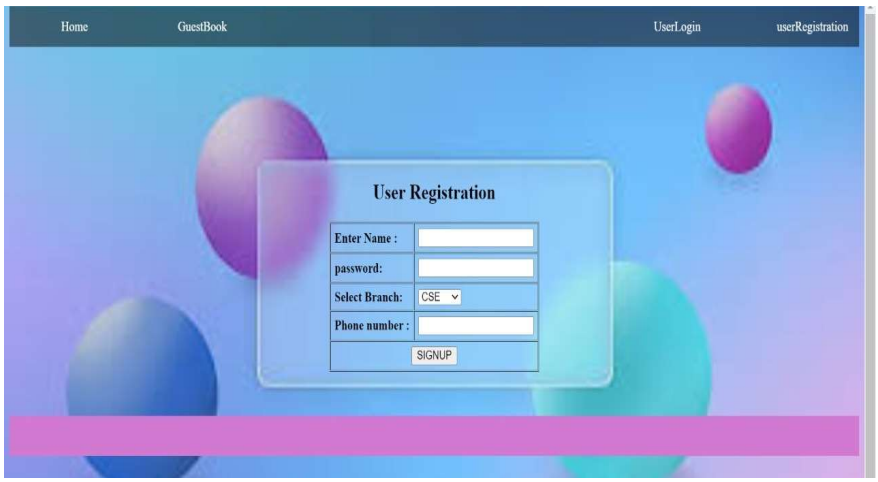
7.1 Steps to give and review the Feedback :

Step-1: Home page

Fig 7.1 home page

- The figure shows the GuestBook, Whatever the feedback that want to give by the guest that can be done through this guestbook

Step-2: user registration & Login :



7.2 user Registration and Login

- The above image is about User Registration.
- User has to SignUp with their basic details like Name, Password and with their Phone number.

Step-3: Guest Registration and Login

fig 7.3 Guest registration and Login

- The above image is about user Login.
- The user has to login with their basic details like Name, Password and with their phno.
- Similary for the guest also have registarion and login options

Step-4:FeedBack Image

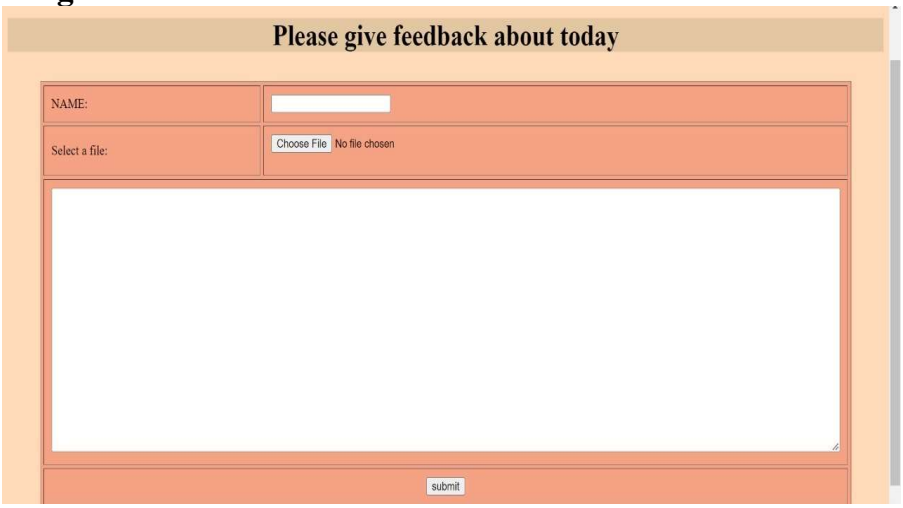


fig 7.4 : Guest feedback upload

- The image is about feedback image.
- Guest has to give feedback in text field area.
- We also choose file, at the time of giving feedback.
- After giving the feedback, click submit button. The data is stored in DataBase.

Step-5: Guest Book page

Fig 7.5 : Guest Book page

- FeedBack given by the Guest is displayed in above image along with choosen image.

Step-6: user Review

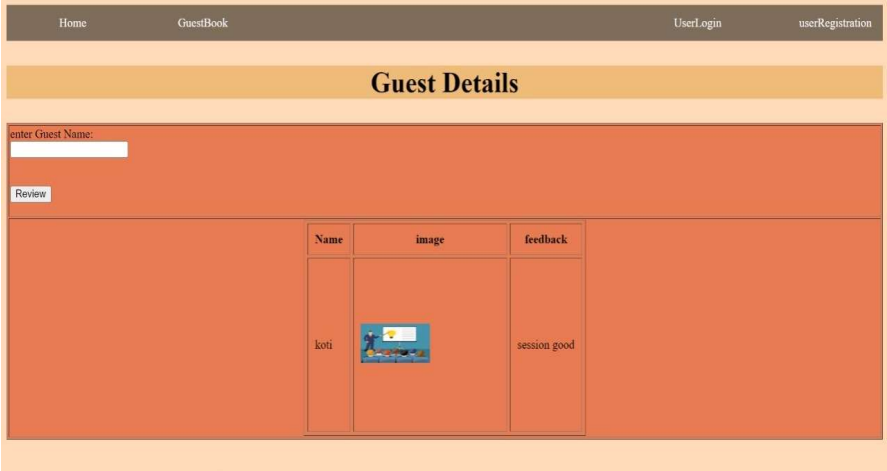


Fig 7.6 user review

- If User wants to know the feedback given by the guest, User has to login with guest details.
- Then it displays the feedback given by the guest.

8.TESTING

8.1 Introduction to testing :

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

- Meets the software and technical requirements that guided its design and development
- Works as expected
- Can be implemented with the same characteristics

8.2 Types of Testing

Acceptance Testing

Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system It is usually performed by the customer.

Active Testing :

Type of testing consisting of introducing test data and analyzing the execution results. It is usually conducted by the testing team

Agile Testing :

Software testing practice that follows the principles of the agile manifesto, emphasizing testing from the perspective of customers who will utilize the system it is usually performed by the QA teams Read More on Agile Testing

Ad-hoc Testing :

Testing performed without planning and documentation - the tester tries to 'break' the system by randomly trying the system's functionality it is performed by the testing team.

Alpha Testing :

Type of testing a software product or system conducted at the developer's site. Usually it is performed by the end users.

Unit Testing :

It focuses on smallest unit of software design. In this we test an individual unit or group of interrelated units. It is often done by programmer by using sample input and observing its corresponding outputs.

Big Bang Integration Testing

Big Bang Integration Testing is an integration testing Strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

User Interface Testing

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output. Integration testing are of two types: (i) Top down (ii) Bottom up.

Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing. It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.

Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment.

System Testing

In this software is tested such that it works fine for different operating system. It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working. In this we have security testing, recovery testing, stress testing and performance testing.

Backend Testing

Backend testing is defined as a type of testing that checks the server side or Database. It is also known as Database Testing. The data entered in the front end will be stored in the back-end database.

8.3 Test Cases and Reports

8.3.1 Test Case:

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of anapplication.

How to write test cases for software:

- Use a Strong Title.
- Include a Strong Description.
- Include Assumptions and Preconditions.
- Keep the Test Steps Clear and Concise.
- Include the Expected result.
- Make it Reusable.

Purpose of test report:

Document that records data obtained from an experiment of Evaluation in an organized manner, describes the environmental or operating conditions, and Shows the comparison of test results with test objectives.

Who prepares test summary report?

Test summary report is a document which contains Summary of test activities and final test results. After the testing cycle it is very important that you communicate the test results and findings to the project stakeholders so that decisions can be made for the software release.

Testcases & reports:

Test case no 1: To check the user registration successful	**Priority:** High
Test objective: Creating account to a user	
Test Description: Uploading required details in the registration portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: Upload the details asked to upload by the user.	**Expected Result:** Showing Registration successful
Pass: Yes **Condition Pass:** Yes **Fail:** No	
Problems or issues: No	
Note: Executed Successfully	

Test case no 2: To check the user login successful	**Priority:** High
Test objective: Logging into user's account	
Test Description: Providing appropriate credentials in login portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: Upload the details asked to upload by the user.	**Expected Result:** User is directed to user homepage
Pass: Yes **Condition Pass:** Yes **Fail:** No	
Problems or issues: Yes	
Note: Executed Successfully	

Test case no 3: To check the guest registration successful	**Priority:** High
Test objective: Creating account to a guest	
Test Description: Uploading required details in the registration portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: Upload the details asked to upload by the user.	**Expected Result:** Showing Registration successful
Pass: Yes **Condition Pass:** Yes **Fail:** No	
Problems or issues: No	
Note: Executed Successfully	

Test case no 4: To check the guest login successful	**Priority:** High
Test objective: Logging into guest's account	
Test Description: Providing appropriate credentials in login portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: Upload the details asked to upload by the guest.	**Expected Result:** guest is directed to user homepage
Pass: Yes **Condition Pass:** Yes **Fail:** No	
Problems or issues: Yes	
Note: Executed Successfully	

Test case no 5: Logging in without registration	**Priority:** Low
Test objective: A new user or guest is logged in without registration	
Test Description: Submitting empty form in submit idea portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: User or guest has to directly login	**Expected Result:** User or guest should not register
Pass: Yes **Condition Pass:** Yes **Fail:** Yes	
Problems or issues: No	
Note: Executed Successfully	

Test case no 6: To check the guest feedback submission successful	**Priority:** High
Test objective: Verifying guest feedback submitted successfully	
Test Description: Uploading required information in idea submit guest information portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: Upload the details asked to upload by the user.	**Expected Result:** Showing feedback successfully .
Pass: Yes **Condition Pass:** Yes **Fail:** No	
Problems or issues: Yes	
Note: Executed Successfully	

Test case no 7: Submitting feedback without filling fields	**Priority:** High
Test objective: Empty feedbacks are submitted or not	
Test Description: Submitting empty form in submit idea portal	
Requirement Verified: Yes	
Test Environment: A pc with internet connection	
Test Setup: System must have an active internet connection	
Actions: Do not upload any details	**Expected Result:** feedback should not be submitted
Pass: Yes **Condition Pass:** Yes **Fail:** Yes	
Problems or issues: Yes	
Note: Executed Successfully	

9.Conclusion & Future Enhancement

9.1 Conclusion :

The prototype model is designed using structured modeling and is able to provide the desired results. It can be successfully implemented as a Real Time system with certain modifications. Science is discovering or creating major breakthrough in various fields, and hence technology keeps changing from time to time.
GuestBook describes a method of collecting the feedback from the guest through online mode. In previous days feedback can be manipulated in the process of manual work, to avoid the drawbacks we proposed a web-based feedback collecting system. The proposed system is very effective ,easy and take less time to organize the accurate feedbacks without human intervention

9.2 Future Enhancement:

In future we will add more features that attracts more users as far as possible.

10.BIBILIOGRAPHY

The following books are referred during the analysis and execution phase of the project

10.1 Reference Books:

1. Jeanine Meyer , "**The guide of html and javascript**"

2. Carolee Cameron **, "web design"**

3.David Flangnan , "*<>*javascript"

4.Laurence Lars Skevis , "Javascript from Beginner to Professional"

5.Dan Bander , "Python tricks:A Buffet Of Awesome Python Features"

6.Laura Cassell , "Python Projects"

7.Taneja Sheetal , "Python Programming|A Modular Approach"

8.Laura Lemay , "Mastering Html/Css & Javascipt Web Publishing"

9.Miguel Grinberg , "Flask Web Development"

10.Michael Herman , "Authentication with Flask,React &Docker""

10.2 Weblinks :

1. https://www.javatpoint.com/html-tutorial
2. https://www.w3schools.com/html/
3. https://www.geeksforgeeks.org/html/
4. https://flask.palletsprojects.com/en/2.2.x/
5. https://www.python.org/