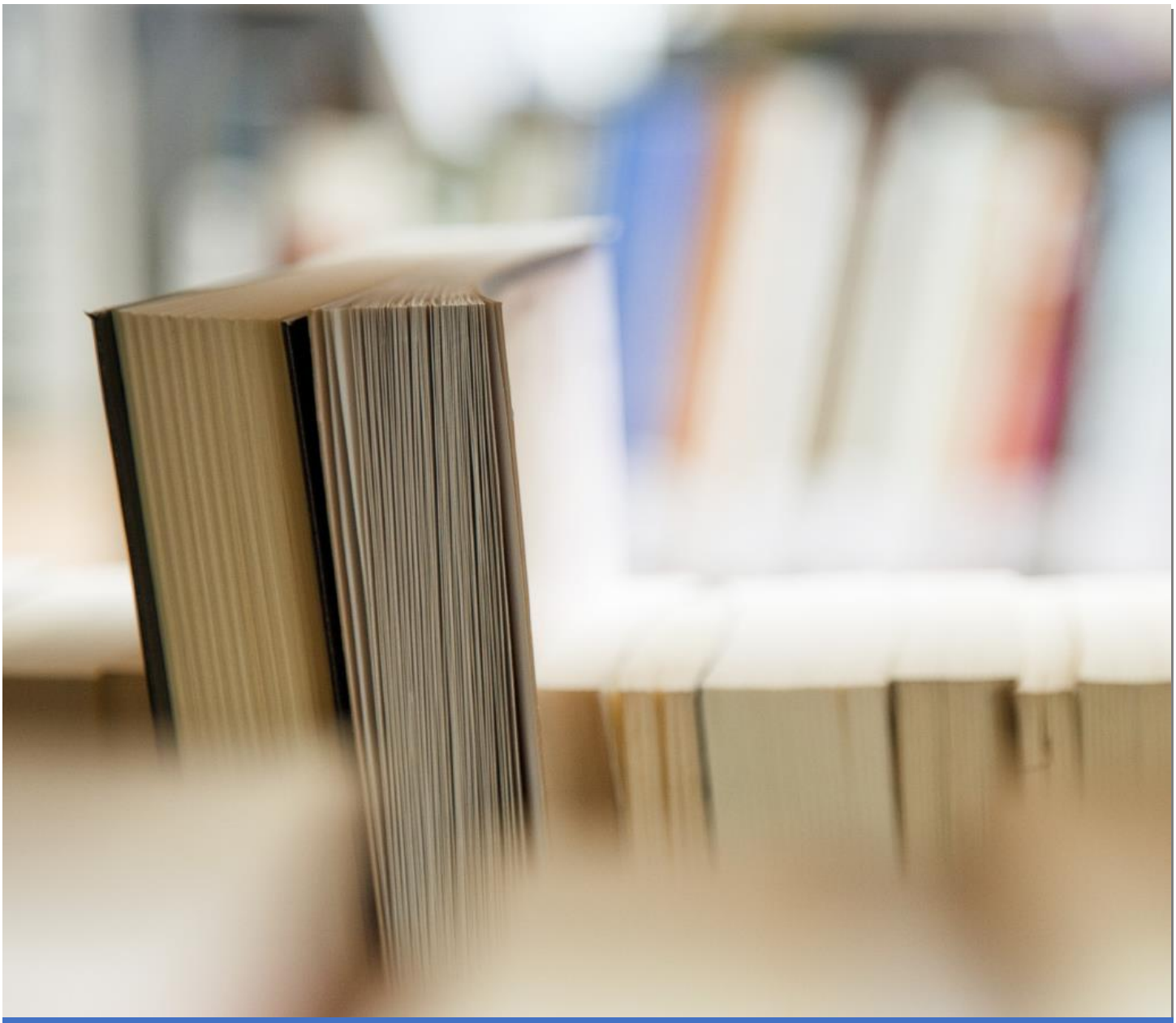


SA4

FEEDBACK FORM APPLICATION

27/06/2019

SA4 INSTITUTIONS

PROJECT REPORT

PROBLEM STATEMENT:

Implement feedback application that includes student portal and trainer portal using flask framework.

Project submitted by

Rachure Akshit

Sai Krishna E L

Amogh J P

Abhishek B B

Arun P N

Trainer Name

Suman Gangopadhyay

Table of contents

Domains

- ✓ Rachure Akshit (1SI17IS030) ……Business-logic
- ✓ Sai Krishna E L (1SI17IS035)…….Business-logic
- ✓ Amogh J P (1SI17IS004)……..……Front-end
- ✓ Abhishek B B (1SI17IS001)……..…Front-end
- ✓ Arun P N (1SI17IS007)……..……Database

Overview of the project

This project implements a feedback application that includes two portals named Trainer portal and Student portal, where trainer portal gets into trainer login page and Student portal gets into feedback page.

- Dash board present at the First/index webpage presents the information about the number of trainers, number of courses available, number of students and the number of feedbacks received and it includes the rating of each trainer.

- The trainer login page contains two fields, trainer name and password, after authentication, feedbacks for his training are shown.

- The Student portal gets to a button which takes to feedback form which has to be filled by the student.

- After satisfying all the conditions applied, the data entered by the student is saved and stored successfully.

- For each feedback received, the rating for that respective trainer is calculated based on rating received in this feedback along with the previous rating and the final rating is updated successfully.

- The student can submit another response on clicking a link present at thank you page.

Requirements

i) Laptop

ii) Internet connection

iii) Text editors
   (a) Visual studio code
   (b) Atom
   (c) Sublime text

iv) Websites
   (a) w3 schools
   (b) stackoverflow
   (c) getbootstrap
   (d) sit.ac.in

v) Database service provider
   (a) Xampp

vi) Database
   (a) Mysql

vii) Backend framework
   (a) Flask

viii) Programming Language
   (a) Python

ix) Frontend
   (a) HTML
   (b) CSS
   (c) JS

FUNCTIONALITIES

- ✓ app.config['SQLALCHEMY_DATABASE_URI']='mysql+pymysql://root:@localhost/flask_project'
  db=SQLAlchemy(app)

  Here, we used mysql database and connect to our piece of code using SQLAlchemy class. db is an instance of class SQLAlchemy and app is the instance of flask class and SQLAlchemy inherits the flask class.

- ✓ class CreateTable(db.Model):
  __tablename__="feedback_table"
  id=db.Column(db.Integer,primary_key=True)
  name=db.Column(db.String(20))
  usn=db.Column(db.String(20))
  ……

  Here, we created the table using ORM(Object Relational Mapper) class named CreateTable() using the instance db.Model. Table name is “feedback_tabel” which includes id (datatype – integer, primary_autoincrement_key), name (datatype – string), usn (datatype – string), etc…

- ✓ class TrainerTable(db.Model):
  __tablename__ = “trainer_table”
  ID = db.Column(db.Integer,primary_key=True)
  Akshit = db.Column(db.Float)
  SaiKrishna = db.Column(db.Float)
  ……..

Here, we created one more table using ORM(Object Relational Mapper) class named TrainerTable() using the instance db.Model. Table name is “trainer_table” which includes ID (datatype – integer, primary_autoincrement_key), Akshit (datatype – float), SaiKrishna (datatype – float), etc…

- ✓
```
data = TrainerTable.query.all()
data1 = CreateTable.query.all()
```

‘data’ holds all the “trainer_table” details using TrainerTable.query.all() model and ‘data1’ holds all the “feedback_table” details using CreateTable.query.all() model.

- ✓
```
@app.route('/index/')
def index():
    return render_template("indexpage.html")
```

This app.route() executes when the url contains ‘127.0.0.1:8080/index/’ and opens the ‘indexpage.html’ page.

- ✓
```
form = request.form
name =  form['name']
usn = form['usn']
phone = form['phn']
mail = form['mail']
college = form['clgname']
…….
```

Here, we are reading the values, that the user (student) gives his/her feedback in the feedback page, to the particular variables by using form = request.form and then these are used to insert the data into the table “feedback_table”.

- ✓ insert=CreateTable(name=name,usn=usn,phn_no=phone,email=mail,college=college,trainer=trainer,course=course,overall_rate=o_rate,hands_on=h_on,pace=pace,explanation=explanation,doubt_clearing=doubt,suggestions=tarea)

Here, we are inserting the variables, that we used to store the response of user (student), to the “feedback_table” using the class name CreateTable() and the whole data is stored in the variable ‘insert’.

- ✓ save=db.session

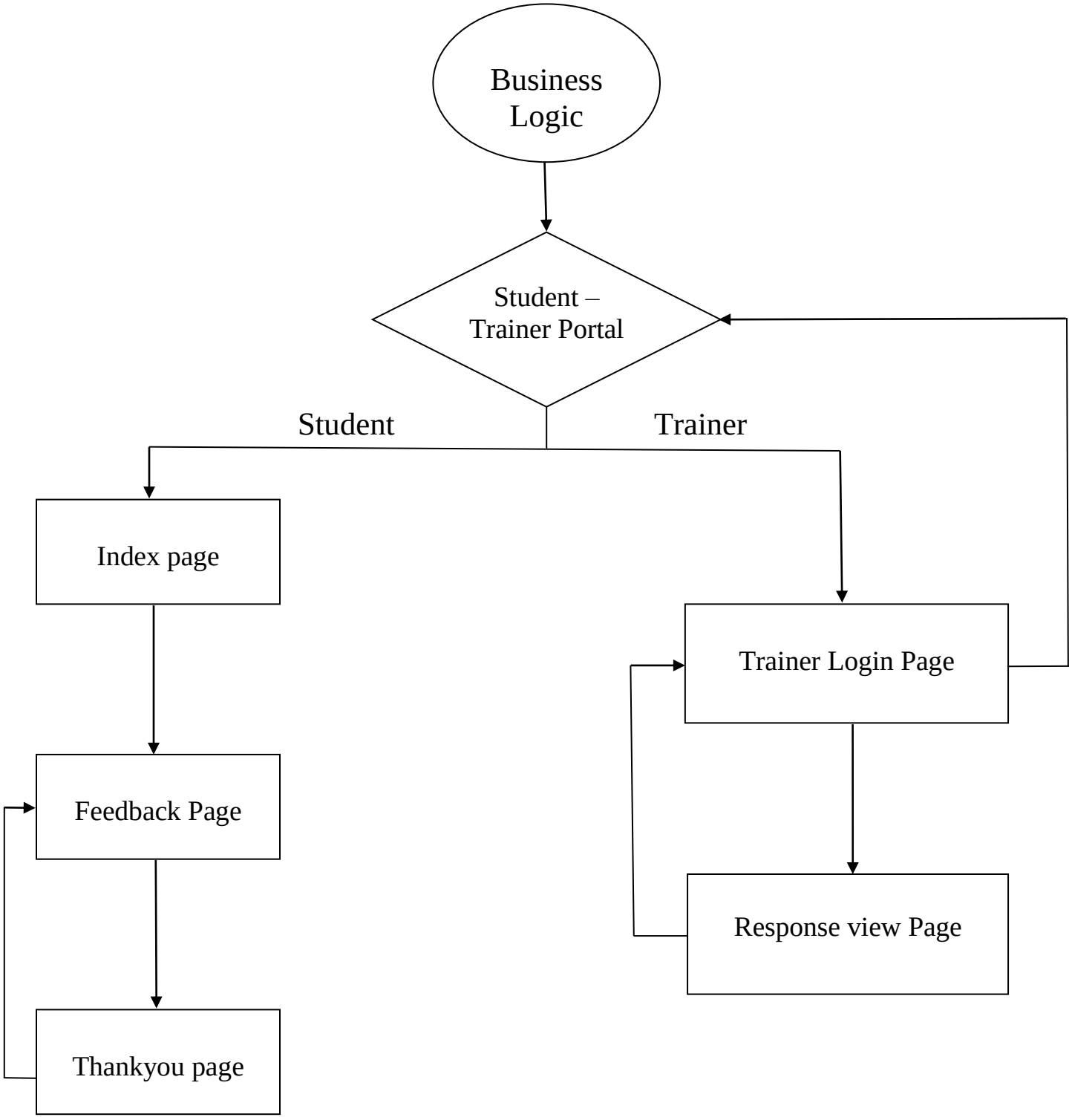
```
        try:
            save.add(insert)
            save.commit()
            return redirect(url_for('thankyou'))
        except Exception as e:
            save.rollback()
            save.flush()
            print("Error in entering data into Database")
            print(e)
    return render_template("feedback123.html")
```

Here, we are creating a session to insert the data into the “feedback_table” by using the db.session. There are two cases,

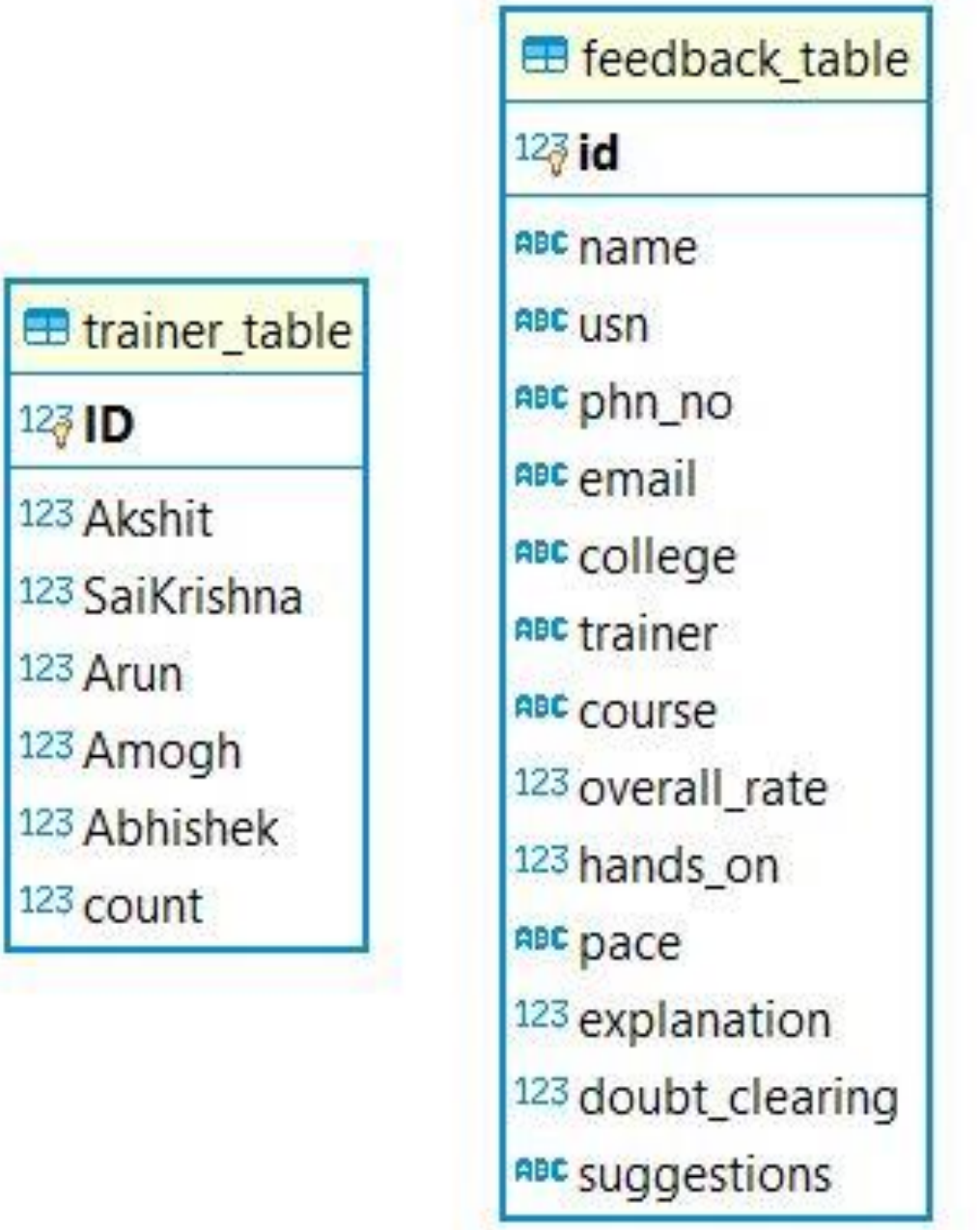
Case 1 : If session is created successfully, then we add the data stored in the variable ‘insert’ and commit it finally we redirect it to ‘thankyou’ html page.

Case 2 : If session is not created, then we will rollback the stored data and flush it finally we return to ‘feedback123’ html page.

Flow of Logic



E R Diagram


feedback_table
id
name
usn
phn_no
email
college
trainer
course
overall_rate
hands_on
pace
explanation
doubt_clearing
suggestions
trainer_table
ID
Akshit
SaiKrishna
Arun
Amogh
Abhishek
count

Development Methodology

The main idea behind this project is to ensure the proper maintenance of an organization, and to change their style of working/training based on the feedbacks received by the candidates appeared for training. This is not just a feedback form but also contains a trainer's portal which takes to the page that displays feedbacks received particularly for that trainer. Those feedbacks contains ratings for each particular domain and suggestions too. So that the trainer can analyze the ratings received and can concentrate/improve in particular domain. To submit multiple responses/feedbacks, A link to submit another response is placed at the end (thank you) page.

Roles and Responsibilities

- **Front-end developer :** To design the html page which is the skeleton of the application and then design that html page using Javascript , css and bootstrap.

- **Back-end developer/ Business logic developer :** To develop the business logic using Python programming language and Flask Framework so that the project operates successfully.

- **Database operator :** To store the data/response entered by the student and fetch the data stored from the database when trainer logs in using mysql database and xampp database server.

Folders

- Static
  - CSS
    - trainer.css
    - bootstrap.min.css
  - img
    - abhi.jpg
    - akshit.jpg
    - Amogh.jpg
    - arun.jpg
    - sai.jpg
    - background.jpg
    - trainer.jpg
- templates
  - feedback123.html
  - indexpage.html
  - S_T.html
  - thankyou.html
  - trainerlogin.html
  - viewall.html
- app.py

*****