

## **DBMS Assignment 4**

### **PESUVariance - Learn, Grow & Challenge Yourself!**

**Team ID: H1**

**Names and respective SRN's:**

**Shrikar Madhu: PES1UG19CS470**

**Sravya Yepuri: PES1UG19CS502**

**Sri Ramya Priya Vedula: PES1UG19CS504**

**Yousha Mahamuni: PES2UG19CS468**

**Semester: 5, Section: H**

#### **Dependencies:**

Backend:

- 1) **Flask:** Flask is a web development framework developed in Python. It is easy to learn and use. Flask is “beginner-friendly” because it does not have boilerplate code or dependencies, which can distract from the primary function of an application. Flask provides a development server and a debugger. Many extensions are available for Flask, which can be used to enhance its functionalities.
- 2) **PostgreSQL:** PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance.
- 3) **Psycopg2:** Psycopg is a PostgreSQL adapter for the Python programming language. This tool allows us to connect the capabilities of the Python language and libraries to obtain, manipulate, input, and update data stored in a PostgreSQL database.

Frontend:

**HTML, CSS, Javascript:** HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

## Screenshots for the statements executed from the front end:

### 1. Register New Account

Register New user or login existing account

Your Name: Jane John

Your Email: jj@gmail.com

Enter User Type: Student

Your Date of Birth: 01-01-2004

Your Country Code: 080

Your Phone Number: 1111111111

Username: jj

Password:

**Register** **Login**

Register New user or login existing account

Your Name:

Your Email:

Enter User Type:

Your Date of Birth:

Your Country Code:

Your Phone Number:

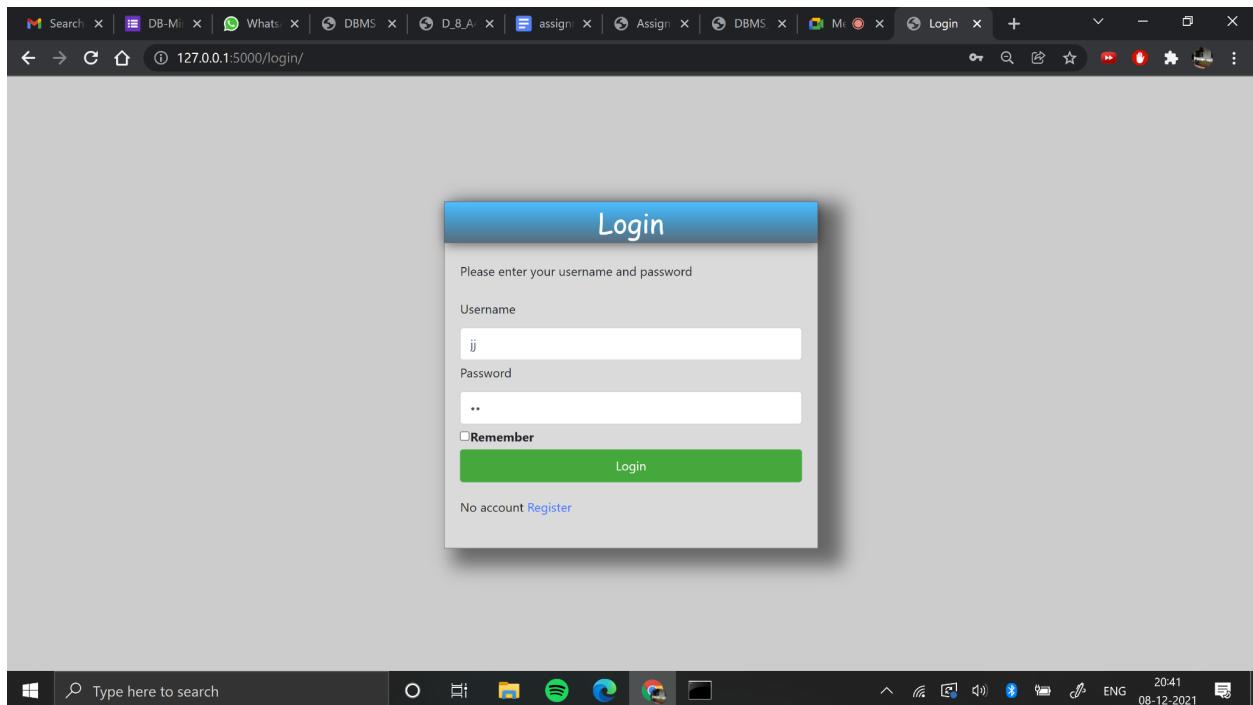
Username:

Password:

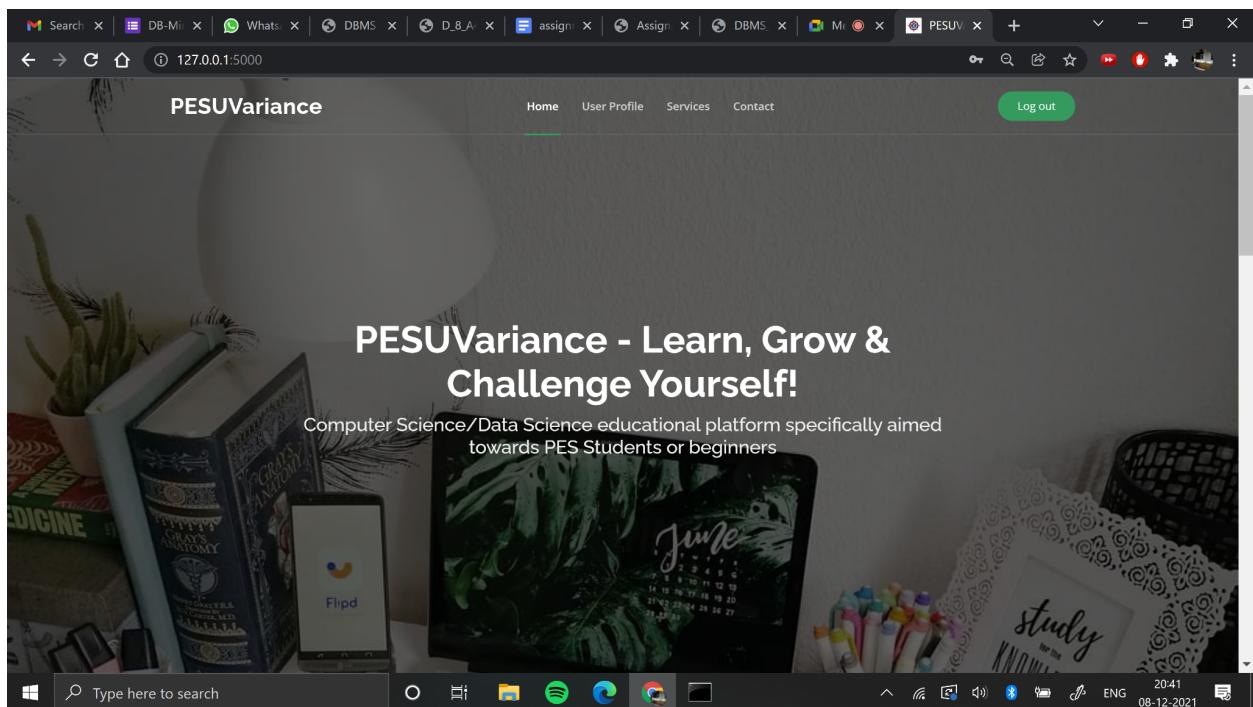
You have successfully registered! X

**Register** **Login**

2. After registering new account:



3. Home page



#### 4. Choose quizzes or competitions

The screenshot shows the 'Services' section of the PESUVariance website. At the top, there's a navigation bar with links for Home, User Profile, Services, and Contact, along with a Log out button. Below the navigation, the page title 'Services' is displayed. A sub-header states: 'We provide many fun quizzes and competitions to test your knowledge with.' Two main sections are shown: 'Quizzes' (with an icon of a document) and 'Competitions' (with an icon of a trophy). Each section has a brief description: 'Click and select which quiz you wanna participate in' for Quizzes and 'Click and select which competition you wanna participate in' for Competitions.

#### 5. Contact us

The screenshot shows the 'Contact' section of the PESUVariance website. At the top, there's a navigation bar with links for Home, User Profile, Services, and Contact, along with a Log out button. Below the navigation, the page title 'Contact' is displayed. A sub-header says: 'Contact us for any grievances'. To the left, there's a map of the Banashankari area in Bengaluru, showing the location of PES University and other nearby landmarks. To the right of the map, there's a form for contacting the university. The form includes fields for 'Your Name', 'Your Email', and 'Subject', each with a corresponding input field. At the bottom of the contact section, there's a note: 'Keyboard shortcuts Map data ©2021 Google Terms of Use'.

## 6. Contact form

The screenshot shows a contact form on a web browser. The header includes a logo for "PESUVariance" and navigation links for Home, User Profile, Services, and Contact. A "Log out" button is also present. The contact form itself has fields for "Your Name", "Your Email", "Subject", and "Message". To the left of the form, there are three sections: "Location" (100 Feet Ring Road, Banashankari Stage III, Dwaraka Nagar, Banashankari, Bengaluru, Karnataka 560085), "Email" (sravya.bg@gmail.com), and "Call" (1234567890). A green "Send Message" button is located at the bottom right of the form area.

## 7. User Profile

The screenshot shows a user profile page on a web browser. The header includes a logo for "PESUVariance" and navigation links for Home, Profile, and Logout. The main content is titled "Profile Page" and displays account details: Name: Jane John, Email ID: jj@gmail.com, Type: Student, Date Of Birth: 2004-01-01, Country Code: 80, and Number: 1111111111. Below this, there is a note: "Your account details are below:". The page is set against a dark background with white text and features a standard Windows taskbar at the bottom.

## Screenshots for the schema change statements:

Added Score Column to participates\_in table.

```
pesuvariance=# \d participates_in
      Table "public.participates_in"
 Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 pb_userid | character varying(5) |           | not null |
 pb_quizid | character varying(5) |           | not null |
Indexes:
 "participates_in_pkey" PRIMARY KEY, btree (pb_userid, pb_quizid)
Foreign-key constraints:
 "participates_in_pb_quizid_fkey" FOREIGN KEY (pb_quizid) REFERENCES quiz(quiz_id)
 "participates_in_pb_userid_fkey" FOREIGN KEY (pb_userid) REFERENCES user_profile(user_id)

pesuvariance=# ALTER TABLE participates_in ADD COLUMN Score NUMERIC(3);
ALTER TABLE

pesuvariance=# \d participates_in
      Table "public.participates_in"
 Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 pb_userid | character varying(5) |           | not null |
 pb_quizid | character varying(5) |           | not null |
 score    | numeric(3,0)        |           |           |
Indexes:
 "participates_in_pkey" PRIMARY KEY, btree (pb_userid, pb_quizid)
Foreign-key constraints:
 "participates_in_pb_quizid_fkey" FOREIGN KEY (pb_quizid) REFERENCES quiz(quiz_id)
 "participates_in_pb_userid_fkey" FOREIGN KEY (pb_userid) REFERENCES user_profile(user_id)
```

## PERFORMANCE ANALYSIS:

```
pesuvariance=# EXPLAIN ANALYZE SELECT * FROM login;
               QUERY PLAN
-----
Seq Scan on login  (cost=0.00..12.50 rows=250 width=292) (actual time=0.008..0.009 rows=8 loops=1)
Planning Time: 0.452 ms
Execution Time: 0.017 ms
(3 rows)

pesuvariance=# EXPLAIN ANALYZE SELECT user_id, activity FROM user_profile WHERE user_id = 'S1';
               QUERY PLAN
-----
Index Scan using user_profile_pkey on user_profile  (cost=0.14..8.16 rows=1 width=56) (actual time=0.033..0.034 rows=1 loops=1)
 Index Cond: ((user_id)::text = 'S1)::text
Planning Time: 0.110 ms
Execution Time: 0.056 ms
(4 rows)
```

```

pesuvariance=# EXPLAIN ANALYZE SELECT Question_Desc, A_Answer_1, A_Answer_2, A_Answer_3, A_Answer_Key FROM QUESTIONS FULL
OUTER JOIN QUIZ ON Q_Quiz_ID = Quiz_ID FULL OUTER JOIN ANSWERS ON A_Question_ID = Question_ID WHERE Quiz_Name = 'Statistics' AND Question_Desc IS NOT NULL;
                                         QUERY PLAN
-----
Nested Loop Left Join  (cost=15.92..29.53 rows=1 width=896) (actual time=0.157..0.189 rows=8 loops=1)
-> Hash Join  (cost=15.78..29.19 rows=1 width=242) (actual time=0.096..0.104 rows=8 loops=1)
    Hash Cond: ((questions.q_quiz_id)::text = (quiz.quiz_id)::text)
        -> Seq Scan on questions  (cost=0.00..12.70 rows=269 width=266) (actual time=0.062..0.064 rows=8 loops=1)
            Filter: (question_desc IS NOT NULL)
        -> Hash  (cost=15.75..15.75 rows=2 width=24) (actual time=0.018..0.019 rows=1 loops=1)
            Buckets: 1024  Batches: 1  Memory Usage: 9kB
                -> Seq Scan on quiz  (cost=0.00..15.75 rows=2 width=24) (actual time=0.009..0.009 rows=1 loops=1)
                    Filter: ((quiz_name)::text = 'Statistics'::text)
                    Rows Removed by Filter: 7
-> Index Scan using answers_pkey on answers  (cost=0.14..0.34 rows=1 width=702) (actual time=0.009..0.009 rows=1 loops=8)
    Index Cond: ((a_question_id)::text = (questions.question_id)::text)
Planning Time: 1.435 ms
Execution Time: 0.252 ms
(14 rows)

```

## **Changes in Business/Application changes/expansion - that might lead to schema changes, constraint changes, DBMS migration (from SQL based to No-SQL)**

In situations where we want to change the data types of features, migrating to a No-SQL would be more efficient. No-SQL databases are better for representing hierarchical relationships between entities.

**With the existing design of your database, if you have to migrate to any No-SQL variety, then which one will be your choice? Why? ( Out of the 4 major varieties, you have to pick one and justify your choice. If you can give comparative features across different flavors, that would be great.)**

We think migrating to a document database like Mongo-DB would be a suitable option as the features of \Mongo-DB are:

1. MongoDB supports semi structured data
2. Indexing: In MongoDB database, every field in the documents is indexed with primary and secondary indices; this makes it easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then the database searches each document with the specified query which takes lots of time and is not so efficient.
3. Powerful querying and analytics: The MongoDB Query Language (MQL) is a full featured, powerful language that allows you to query deep into documents, and even perform complex analytics pipelines with just a few lines of JSON-like MQL.
4. Document Oriented: In MongoDB, all the data is stored in the documents instead of tables like in RDBMS. In these documents, the data is stored in fields(key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.

**Contribution of each member:**

Shrikar Madhu - frontend, backend  
Sravya Yepuri - frontend, backend  
Sri Ramya Priya Vedula - frontend, backend  
Yousha Mahamuni - frontend, backend

**Time spent to complete assignment 4:**

Frontend - 8 Hours  
Backend - 8 Hours  
Making final report - 2 Hours