

SPRINT-3

| | |
|--------------|--------------------------|
| Date | 14/11/2022 |
| Team ID | PNT2022TMID07458 |
| Project Name | Plasma Donor Application |

Fill up the Plasma Donor Details :

PLASMA DONOR

NAME
Samyuktha

EMAIL
gusamyuktha@gmail.com

PHONE
+919025041365

BLOOD GROUP
O+

LAST DONATED DATE
06-02-2022

ADDRESS
abcd

DISTRICT

Sanyuktha

EMAIL

gusanyuktha@gmail.com

PHONE

+919025041365

BLOOD GROUP

O+

LAST DONATED DATE

08-02-2022

ADDRESS

#bod

DISTRICT

Mayiladuthurai

STATE

Tamilnadu

AGE

20

SUBMIT

Coding for Plasma Donation :

```

90     return render_template("request.html", msg="Data saved successfully")
91
92
93 @app.route('/donorform', methods = ['POST', 'GET'])
94 def donorform():
95     if request.method == 'POST':
96
97         name = request.form['name']
98         email = request.form['email']
99         phone = request.form['phone']
100        bloodgroup = request.form['bloodgroup']
101        date = request.form['date']
102        address = request.form['address']
103        district = request.form['district']
104        state = request.form['state']
105        age = request.form['age']
106
107        insert_sql = "INSERT INTO plasmadonate VALUES (?, ?, ?, ?, ?, ?, ?, ?)"
108        prep_stmt = ibm_db.prepare(conn, insert_sql)
109        ibm_db.bind_param(prepare_stmt, 1, name)
110        ibm_db.bind_param(prepare_stmt, 2, email)
111        ibm_db.bind_param(prepare_stmt, 3, phone)
112        ibm_db.bind_param(prepare_stmt, 4, bloodgroup)
113        ibm_db.bind_param(prepare_stmt, 5, date)
114        ibm_db.bind_param(prepare_stmt, 6, address)
115        ibm_db.bind_param(prepare_stmt, 7, district)
116        ibm_db.bind_param(prepare_stmt, 8, state)
117        ibm_db.bind_param(prepare_stmt, 9, age)
118        ibm_db.execute(prepare_stmt)
119
120

```

Store the plasma donor details to the database :

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

TQJ08800.PLASMADONATE Back

Export to CSV

| NAME | EMAIL | PHONE | BLOODGROUP | LASTDATE | ADDRESS | DISTRICT | STATE | AGE |
|-----------|-----------------------|--------------|------------|------------|---------|----------------|-----------|-----|
| Samyuktha | gusamyuktha@gmail.com | 919025041365 | O+ | 2022-02-06 | abcd | Mayiladuthurai | Tamilnadu | 20 |

Plasma Receipient fill the form :

HOME ABOUT LOGIN REQUEST DONATE Search

PLASMA RECEIPT

NAME
ABC

EMAIL
abc@gmail.com

PHONE
1234567890

BLOOD GROUP
A-

DATE
10-11-2022

ADDRESS
423

DISTRICT
Madurai

STATE

NAME: ABC
 EMAIL: abc@gmail.com
 PHONE: 1234565434
 BLOOD GROUP: A-
 DATE: 10-11-2022
 ADDRESS: 423
 DISTRICT: Madurai
 STATE: Tamilnadu
 AGE: 20
 SUBMIT

Data Saved Successfully :

HOME ABOUT LOGIN REQUEST DONATE
 Type To text Search

Data saved successfully
PLASMA RECEIPT
 NAME: Enter your name
 EMAIL: Enter your email
 PHONE: Enter your phone number
 BLOOD GROUP: Enter your blood group
 DATE: dd-mm-yyyy
 ADDRESS:
 DISTRICT: Enter your district here

Plasma Request Code :

The screenshot shows the Visual Studio Code editor with a Flask application. The Explorer sidebar on the left shows a project structure with files like `home.html`, `donor.html`, `request.html`, and `app.py`. The main editor displays the `app.py` file, which contains a Flask route `plasmareq` that handles POST and GET requests. The route uses `request.form` to capture form data and inserts it into a database table named `plasmarequest` using a prepared statement. The database connection is established using `ibm_db`. The status bar at the bottom indicates the file is at line 116, column 43, and the Python version is 3.9.7.

```
63 @app.route('/plasmareq', methods = ['POST', 'GET'])
64 def plasmareq():
65     if request.method == 'POST':
66
67         name = request.form['name']
68         email = request.form['email']
69         phone = request.form['phone']
70         bloodgroup = request.form['bloodgroup']
71         date = request.form['date']
72         address = request.form['address']
73         district = request.form['district']
74         state = request.form['state']
75         age = request.form['age']
76
77         insert_sql = "INSERT INTO plasmarequest VALUES (?, ?, ?, ?, ?, ?, ?, ?)"
78         prep_stmt = ibm_db.prepare(conn, insert_sql)
79         ibm_db.bind_param(prepare_stmt, 1, name)
80         ibm_db.bind_param(prepare_stmt, 2, email)
81         ibm_db.bind_param(prepare_stmt, 3, phone)
82         ibm_db.bind_param(prepare_stmt, 4, bloodgroup)
83         ibm_db.bind_param(prepare_stmt, 5, date)
84         ibm_db.bind_param(prepare_stmt, 6, address)
85         ibm_db.bind_param(prepare_stmt, 7, district)
86         ibm_db.bind_param(prepare_stmt, 8, state)
87         ibm_db.bind_param(prepare_stmt, 9, age)
88         ibm_db.execute(prepare_stmt)
89
90     return render_template('request.html', msg="Data saved successfully")
91
```

Store the plasma Request details to the database :

The screenshot shows a database management tool interface with a table named `TQ308800.PLASMAREQUEST`. The table has columns for `NAME`, `EMAIL`, `PHONE`, `BLOODGROUP`, `DATE`, `ADDRESS`, `DISTRICT`, `STATE`, and `age`. Two rows of data are visible, representing plasma requests from ABC and asdf.

| NAME | EMAIL | PHONE | BLOODGROUP | DATE | ADDRESS | DISTRICT | STATE | age |
|------|---------------|------------|------------|------------|---------|----------|-----------|-----|
| ABC | abc@gmail.com | 1234553434 | A- | 2022-11-10 | 423 | Madurai | Tamilnadu | 20 |
| asdf | abc@gmail.com | 9363189486 | O+ | 1990-07-07 | 123 | MYD | Tamilnadu | 7 |