

ASSIGNMENT-2

PHASE-1

TECHNOLOGY: CLOUD APPLICATION DEVELOPMENT

DOMAIN: RETAILS AND E-COMMERCE

TOPIC: SMART FASHION RECOMMENDER APPICATION

@ @ -0,0 +1,4 @ @

1. Create User table with email, username, roll number, password.
 2. Perform UPDATE, DELETE Queries with user table
 3. Connect python code to db2.
 4. Create a flask app with registration page, login page and welcome page. By default, load the registration page once the user enters all the fields store the data in database and navigate to login page.
- Authenticate user with username and password. If the user is valid show the welcome page



Limit to 1000 rows

```
1 • CREATE DATABASE IF NOT EXISTS `geeklogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
2 • USE `geeklogin`;
3
4 • CREATE TABLE IF NOT EXISTS `accounts` (
5     `id` int(11) NOT NULL AUTO_INCREMENT,
6     `username` varchar(50) NOT NULL,
7     `password` varchar(255) NOT NULL,
8     `email` varchar(100) NOT NULL,
9     PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
11
12
13
```



Run >

```
* {
  box-sizing: border-box;
  font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto,
oxygen, ubuntu, cantarell, "fira sans", "droid sans", "helvetica neue",
Arial, sans-serif;
  font-size: 16px;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
body {
  background-color: #435165;
  margin: 0;
}
.login, .register {
  width: 400px;
  background-color: #ffffff;
  box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
  margin: 100px auto;
}
.login h1, .register h1 {
  text-align: center;
  color: #5b6574;
  font-size: 24px;
  padding: 20px 0 20px 0;
  border-bottom: 1px solid #dee0e4;
}
```



Run >

```

.login .links, .register .links {
  display: flex;
  padding: 0 15px;
}
.login .links a, .register .links a {
  color: #adb2ba;
  text-decoration: none;
  display: inline-flex;
  padding: 0 10px 10px 10px;
  font-weight: bold;
}
.login .links a:hover, .register .links a:hover {
  color: #9da3ac;
}
.login .links a.active, .register .links a.active {
  border-bottom: 3px solid #3274d6;
  color: #3274d6;
}
.login form, .register form {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  padding-top: 20px;
}
.login form label, .register form label {
```

javascript:void(0) ... flex...



Run >

```
display: flex;
justify-content: center;
align-items: center;
width: 50px;
height: 50px;
background-color: #3274d6;
color: #ffffff;
}
.login form input[type="password"], .login form input[type="text"], .login
form input[type="email"], .register form input[type="password"], .register
form input[type="text"], .register form input[type="email"] {
width: 310px;
height: 50px;
border: 1px solid #dee0e4;
margin-bottom: 20px;
padding: 0 15px;
}
.login form input[type="submit"], .register form input[type="submit"] {
width: 100%;
padding: 15px;
margin-top: 20px;
background-color: #3274d6;
border: 0;
cursor: pointer;
font-weight: bold;
color: #ffffff;
}
```



Run >

```
transition: background-color 0.2s;
}
.login form input[type="submit"]:hover, .register form
input[type="submit"]:hover {
background-color: #2868c7;
transition: background-color 0.2s;
}
.navtop {
background-color: #2f3947;
height: 60px;
width: 100%;
border: 0;
}
.navtop div {
display: flex;
margin: 0 auto;
width: 1000px;
height: 100%;
}
.navtop div h1, .navtop div a {
display: inline-flex;
align-items: center;
}
.navtop div h1 {
flex: 1;
font-size: 24px;
}
```



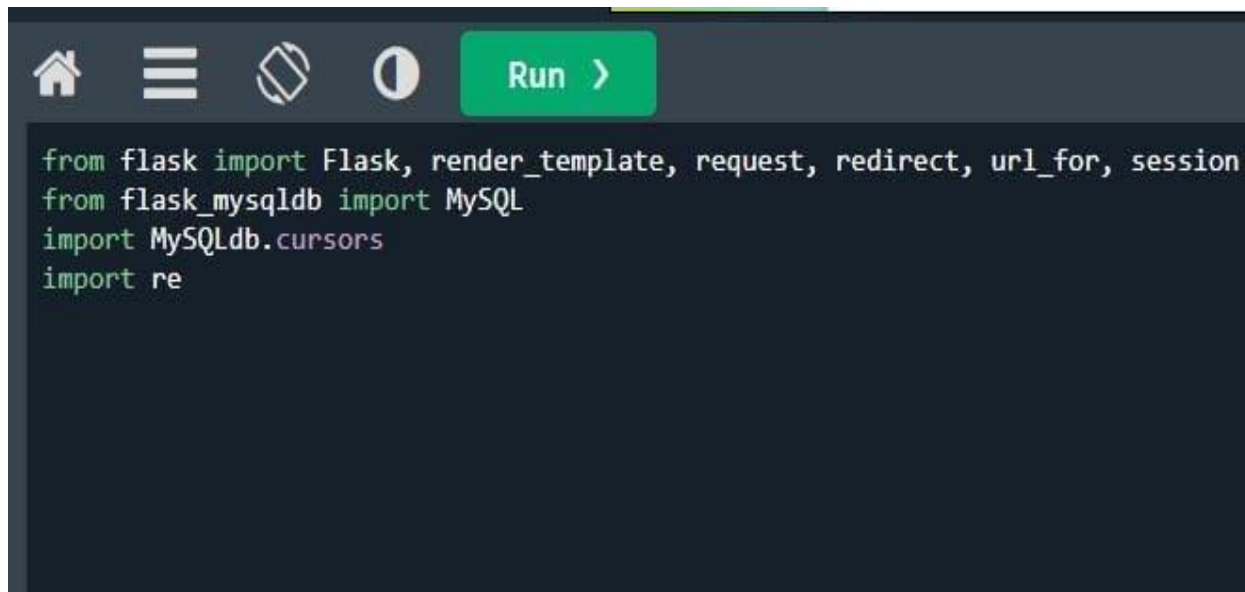
Run >

```
padding: 0;
margin: 0;
color: #eaebed;
font-weight: normal;
}
.navtop div a {
padding: 0 20px;
text-decoration: none;
color: #c1c4c8;
font-weight: bold;
}
.navtop div a i {
padding: 2px 8px 0 0;
}
.navtop div a:hover {
color: #eaebed;
}
body.loggedin {
background-color: #f3f4f7;
}
.content {
width: 1000px;
margin: 0 auto;
}
.content h2 {
margin: 0;
```

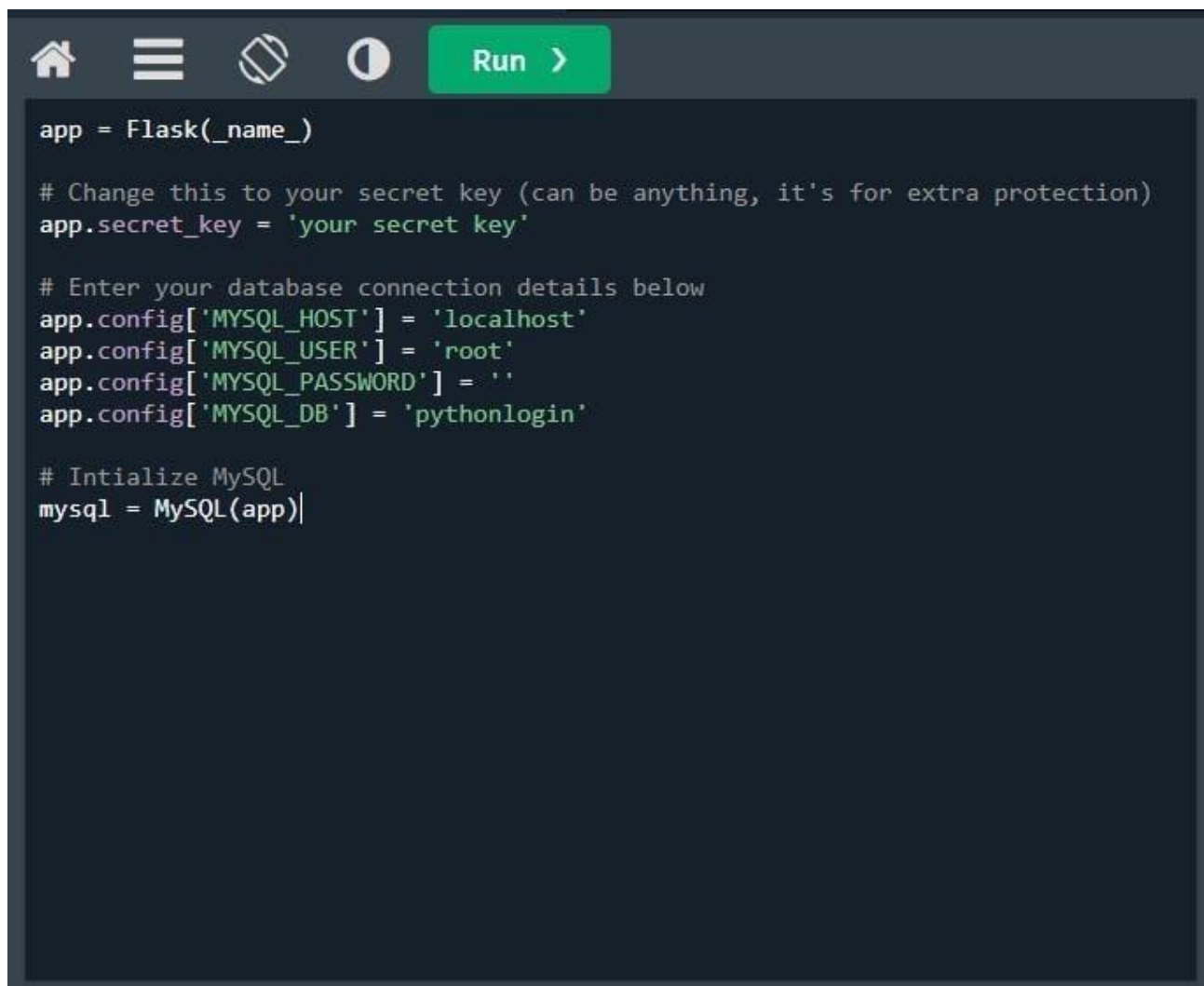


Run >

```
margin: 0;
padding: 25px 0;
font-size: 22px;
border-bottom: 1px solid #e0e0e3;
color: #4a536e;
}
.content > p, .content > div {
box-shadow: 0 0 5px 0 rgba(0, 0, 0, 0.1);
margin: 25px 0;
padding: 25px;
background-color: #fff;
}
.content > p table td, .content > div table td {
padding: 5px;
}
.content > p table td:first-child, .content > div table td:first-child {
font-weight: bold;
color: #4a536e;
padding-right: 15px;
}
.content > div p {
padding: 5px;
margin: 0 0 10px 0;
}
```

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysqlldb import MySQL
import MySQLdb.cursors
import re
```



```
app = Flask(__name__)

# Change this to your secret key (can be anything, it's for extra protection)
app.secret_key = 'your secret key'

# Enter your database connection details below
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'pythonlogin'

# Intialize MySQL
mysql = MySQL(app)
```

```

# http://localhost:5000/pythonlogin/ - the following will be our login page,
which will use both GET and POST requests
@app.route('/pythonlogin/', methods=['GET', 'POST'])
def login():
    # Output message if something goes wrong...
    msg = ''
    return render_template('index.html', msg=msg)|

```

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Login</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
  </head>
  <body>
    <div class="login">
      <h1>Login</h1>
      <div class="links">
        <a href="{{ url_for('login') }}" class="active">Login</a>
        <a href="#">Register</a>
      </div>
      <form action="{{ url_for('login') }}" method="post">
        <label for="username">
          <i class="fas fa-user"></i>
        </label>
        <input type="text" name="username" placeholder="Username"
id="username" required>
        <label for="password">
          <i class="fas fa-lock"></i>
        </label>

```


Login

Login Register



Username




Password

Login




Run >

```
# Check if "username" and "password" POST requests exist (user submitted form)
if request.method == 'POST' and 'username' in request.form and 'password'
in request.form:
    # Create variables for easy access
    username = request.form['username']
    password = request.form['password']
```




```
# If account exists in accounts table in our database
if account:
    # Create session data, we can access this data in other routes
    session['loggedin'] = True
    session['id'] = account['id']
    session['username'] = account['username']
    # Redirect to home page
    return 'Logged in successfully!'
else:
    # Account doesn't exist or username/password incorrect
    msg = 'Incorrect username/password!'
```


javascript:void(0);



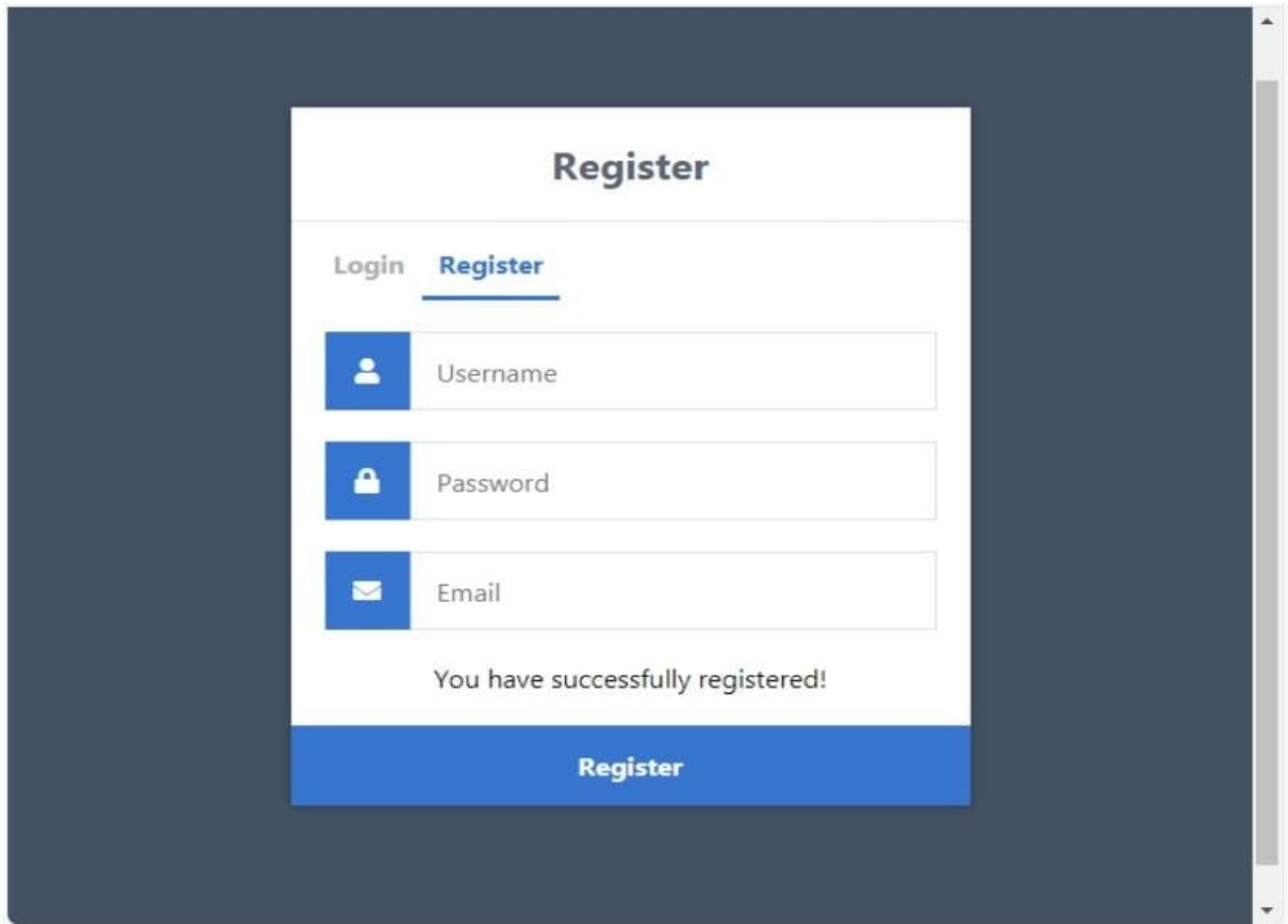
```
@app.route('/pythonlogin/', methods=['GET', 'POST'])
def login():
    # Output message if something goes wrong...
    msg = ''
    # Check if "username" and "password" POST requests exist (user submitted form)
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        # Create variables for easy access
        username = request.form['username']
        password = request.form['password']
        # Check if account exists using MySQL
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM accounts WHERE username = %s AND password = %s', (username, password,))
        # Fetch one record and return result
        account = cursor.fetchone()
        # If account exists in accounts table in our database
        if account:
            # Create session data, we can access this data in other routes
            session['loggedin'] = True
            session['id'] = account['id']
            session['username'] = account['username']
            # Redirect to home page
            return 'Logged in successfully!'
        else:
            # Account doesn't exist or username/password incorrect
            msg = 'Incorrect username/password!'
```



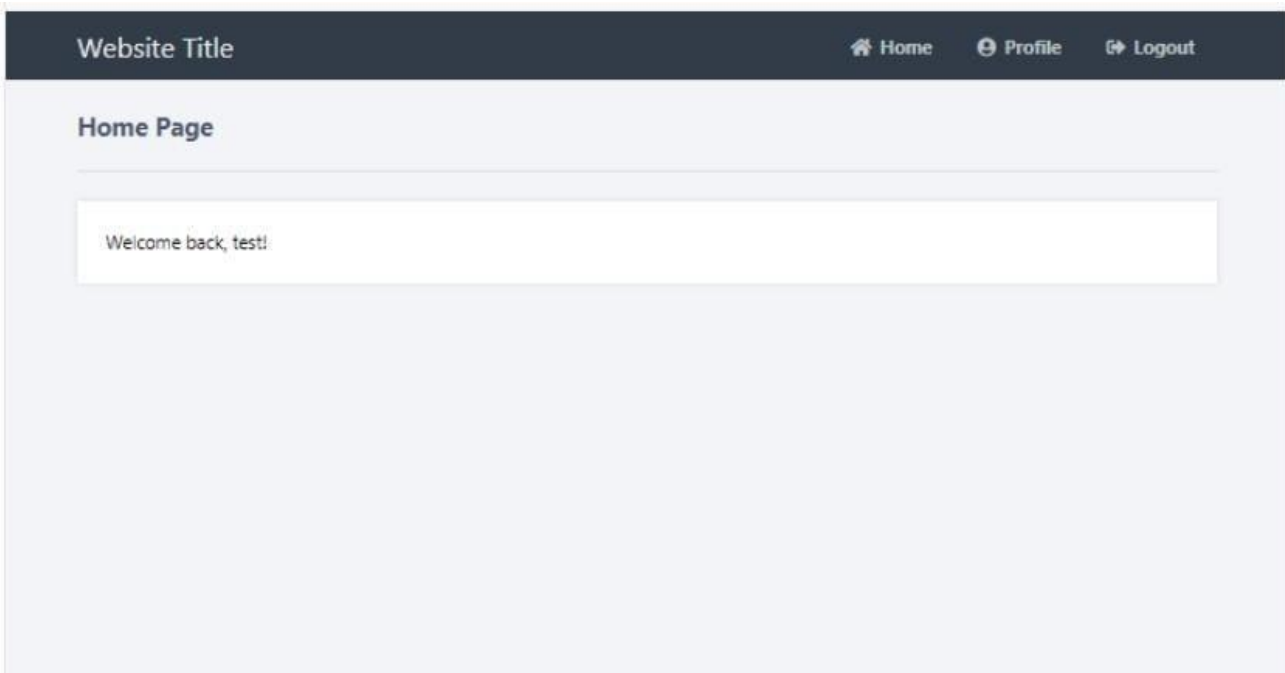
```
# Output message if something goes wrong...
msg = ''
# Check if "username" and "password" POST requests exist (user submitted
form)
if request.method == 'POST' and 'username' in request.form and 'password'
in request.form:
    # Create variables for easy access
    username = request.form['username']
    password = request.form['password']
    # Check if account exists using MySQL
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM accounts WHERE username = %s AND
password = %s', (username, password,))
    # Fetch one record and return result
    account = cursor.fetchone()
    # If account exists in accounts table in our database
    if account:
        # Create session data, we can access this data in other routes
        session['loggedin'] = True
        session['id'] = account['id']
        session['username'] = account['username']
        # Redirect to home page
        return 'Logged in successfully!'
    else:
        # Account doesn't exist or username/password incorrect
        msg = 'Incorrect username/password!'
# Show the login form with message (if any)
return render_template('index.html', msg=msg)
```



```
# http://localhost:5000/pythonlogin/register - this will be the registration
page, we need to use both GET and POST requests
@app.route('/pythonlogin/register', methods=['GET', 'POST'])
def register():
    # Output message if something goes wrong...
    msg = ''
    # Check if "username", "password" and "email" POST requests exist (user
submitted form)
    if request.method == 'POST' and 'username' in request.form and 'password'
in request.form and 'email' in request.form:
        # Create variables for easy access
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
    elif request.method == 'POST':
        # Form is empty... (no POST data)
        msg = 'Please fill out the form!'
    # Show registration form with message (if any)
    return render_template('register.html', msg=msg)|
```



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
  </head>
  <body class="loggedin">
    <nav class="navtop">
      <div>
        <h1>Website Title</h1>
        <a href="{{ url_for('home') }}"><i class="fas fa-home"></i>Home</a>
        <a href="{{ url_for('profile') }}"><i class="fas fa-user-circle">
</i>Profile</a>
        <a href="{{ url_for('logout') }}"><i class="fas fa-sign-out-alt">
</i>Logout</a>
      </div>
    </nav>
    <div class="content">
      {% block content %}{% endblock %}
    </div>
  </body>
</html>
```

```

≡  📄  🔗  ⌚  Run >

{% extends 'layout.html' %}

{% block title %}Profile{% endblock %}

{% block content %}
<h2>Profile Page</h2>
<div>
  <p>Your account details are below:</p>
  <table>
    <tr>
      <td>Username:</td>
      <td>{{ account['username'] }}</td>
    </tr>
    <tr>
      <td>Password:</td>
      <td>{{ account['password'] }}</td>
    </tr>
    <tr>
      <td>Email:</td>
      <td>{{ account['email'] }}</td>
    </tr>
  </table>
</div>
{% endblock %}

```

Profile Page

Your account details are below:

Username: test

Password: test

Email: test@gmail.com

TEAM LEADER: ABINISHA.A

TEAM MEMBERS:

- ABITHA PAULINE.K
- HARI PRIYA.P
- YUVASHREE.S

TEAM ID: PNT2022TMID37964

TEAM SIZE: 4

TEAM MENTOR: SOWJANYA, SANDEEP

TEAM FACULTY: DR S SIVASUBRAMANIAN