Problem Statement: Create a user registration web page such as Name, Email Address and Date of birth. Write a script to calculate age based on the date of birth and store the records.

Project Overview:

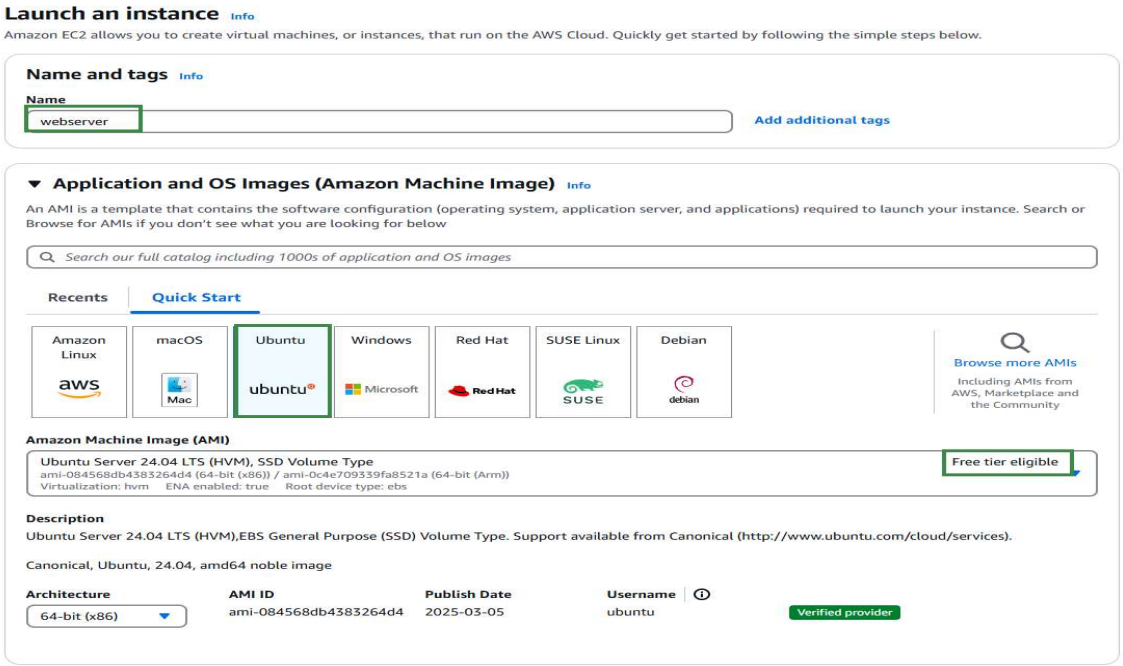
1. A Web application hosted on the EC2 instance to capture user input.
2. A MySQL database hosted on the same EC2 instance to store the records
3. A script to calculate and update the current age based on the date of birth

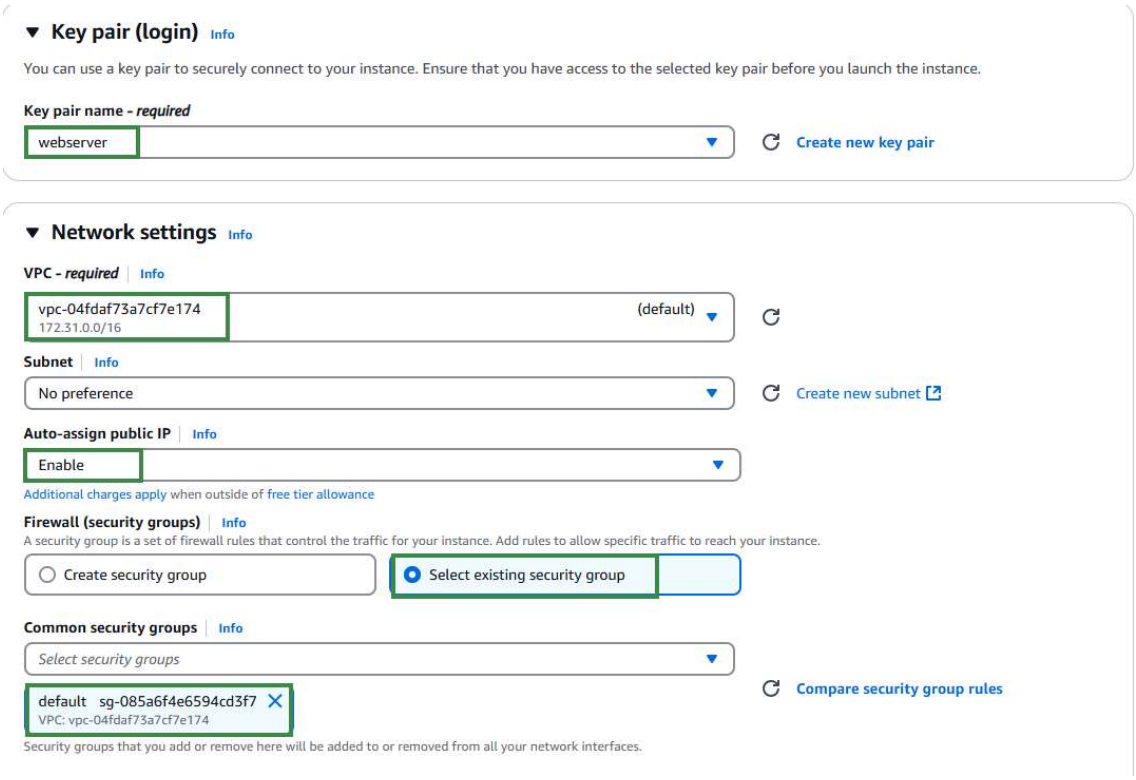
Pre-requisites:

1. Amazon Free Tier Account.
2. Knowledge on how to create IAM users and add permissions.
3. Python or PHP knowledge

Steps:

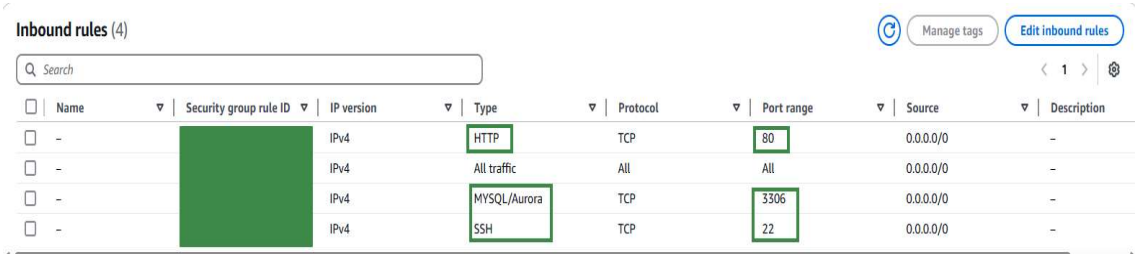
1. Launch an EC2 instance
2. Choose Ubuntu
3. Select Free Tier
4. Security Groups - Allow the following SSH , HTTP and MySQL Ports





5. Leave the other settings as default and click on launch instance
6. After the EC2 instance is launched, perform the following steps
    a. SSH into EC2 using the .pem key
    b. Update the system packages
        i. sudo apt update -y
        ii. Install apache , php and mysql
        iii. sudo apt install -y apache2
        iv. sudo apt install -y php php-mysql mysql-server
        v. start and enable Apache & mysql
            1. sudo systemctl start apache2
            2. sudo systemctl restart apache2
            3. sudo systemctl status mysql
            4. systemctl start mysql

7. In the security group add the following rules as shown
    a. Allow SSH – port 22
    b. Allow HTTP – port 80
    c. Allow mysql – Port 3306



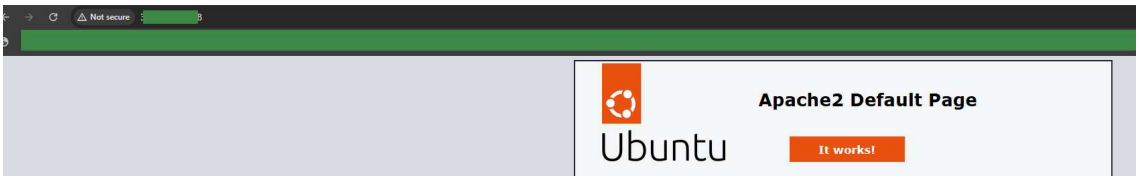
8. Connect to the EC2 instance through SSH from your command terminal
9. Always remember to attach internet gateway to the VPC and update the route so that there is no connection timeout error.

```
ubuntu@ip-[redacted]:~$ sudo apt update -y
```

```
ubuntu@ip-[redacted]:~$ sudo apt install -y apache2
```

```
ubuntu@ip-[redacted]:~$ sudo apt install -y php php-mysql mysql-server
```

10. Both the services are running and enabled. Copy the EC2 public IP and paste it in the browser address bar.



11. Run the command in the terminal and check the status of Apache2 and mysql
    a. sudo service apache2 status
    b. sudo service mysql status

12. Connect to the sql using the following command

```
ubuntu@ip-[redacted]:~$ sudo su
root@ip-172-31-92-235:/home/ubuntu# mysql -u root -p
Enter password:
```

13. Create a new database to store the user inputs. By default there are databases and type the command as shown

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.08 sec)
```

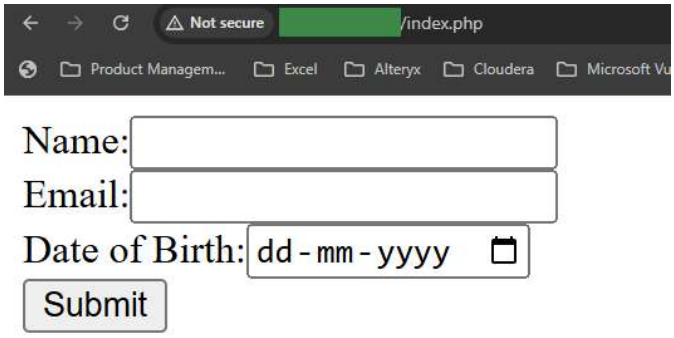
14. Create a database and switch to the database to create a table. Create a table as shown and there will be no records.

```
mysql> CREATE DATABASE user_data;
Query OK, 1 row affected (0.01 sec)

mysql> USE user_data;
Database changed
```

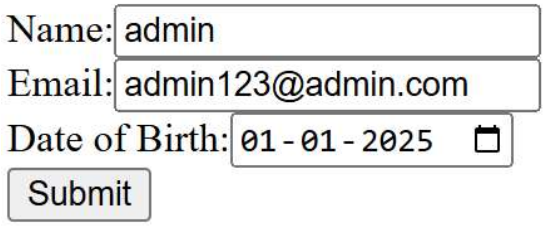
```
mysql> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     email VARCHAR(255) UNIQUE NOT NULL,
    ->     date_of_birth DATE NOT NULL,
    ->     age INT DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> show tables;
+----------------------+
| Tables_in_user_data  |
+----------------------+
| users                |
+----------------------+
1 row in set (0.00 sec)

mysql> select * from users;
Empty set (0.01 sec)

mysql>
```

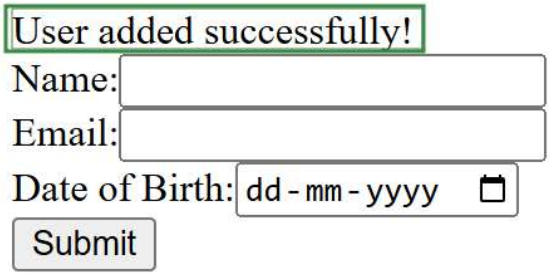
15. The code is written in php and modify the index.html with the php code
16. If you get an error check the index.php code whether the username , password and database name is correct. If you continue to get an error check the apache error log
17. Run the command **sudo tail -n 20 /var/log/apache2/error.log for ubuntu**
18. Sometimes the password might not match or insufficient permissions while using root as the username.
19. **It is best not to use root username to run apps.**
20. Create a new user and update the php code
21. The command to **create a new user is**
    a. CREATE USER 'app'@'localhost' IDENTIFIED BY 'admin@123';
    b. GRANT ALL PRIVILIGES on user_data.* TO 'app'@'localhost';
    c. FLUSH PRIVILEGES;
22. Brief explanation of the above code
    a. Grants all privileges means SELECT, INSERT, UPDATE, DELETE only on the user_data database
    b. User_data.* means all the tables in the user_data database and can manage only one specific database reducing the risk on other databases.
    c. FLUSH means reload the user privilege table immediately
23. Copy the EC2 IP and paste it in the browser. Append /index



24. Insert a record and click on submit



25. The record is inserted successfully.



26. Check the tables if the record is insert successfully

```
mysql> select * from users;
+----+-------+-------------------+---------------+------+
| id | name  | email             | date_of_birth | age  |
+----+-------+-------------------+---------------+------+
|  1 | admin | admin123@admin.com | 2025-01-01    | NULL |
+----+-------+-------------------+---------------+------+
1 row in set (0.00 sec)
```

27. The age is not calculated. Create a script that can be setup as a cron job to run. For the demo purpose , will execute in the terminal
28. Create another script and name it as update_age.php
29. Save the script in the same path as index.php

```
root@ip-[redacted]:/var/www/html# ls -lsa
total 20
4 drwxr-xr-x 2 root     root     4096 Mar 22 15:29 .
4 drwxr-xr-x 3 root     root     4096 Mar 22 11:50 ..
4 -rw-r--r-- 1 www-data www-data  888 Mar 22 12:51 index.php
4 -rw-r--r-- 1 root     root       20 Mar 22 12:44 [redacted]
4 -rw-r--r-- 1 root     root      598 Mar 22 15:29 update_age.php
root@ip-172-31-92-235:/var/www/html#
```

```
root@ip-[redacted]:/var/www/html# php -v
PHP 8.3.6 (cli) (built: Dec  2 2024 12:36:18) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
root@ip-[redacted]:/var/www/html# php update_age.php
Age updated successfully root@ip-[redacted]:/var/www/html#
```

```
mysql> select * from users;
+----+-----------+--------------------+---------------+------+
| id | name      | email              | date_of_birth | age  |
+----+-----------+--------------------+---------------+------+
|  1 | admin     | admin123@admin.com | 2025-01-01    |    0 |
|  4 | tester    | tester@tester.com  | 2025-03-01    |    0 |
|  5 | Test User | test@example.com   | 1990-01-01    |   35 |
|  6 | test1     | test1@test1.com    | 1999-09-02    |   25 |
|  7 | test2     | test2@test2.com    | 2025-02-01    |    0 |
|  8 | test3     | test3@test3.com    | 2024-12-31    |    0 |
|  9 | test4     | test4@test4.com    | 2024-12-31    |    0 |
+----+-----------+--------------------+---------------+------+
7 rows in set (0.00 sec)
```

In the age you can see that for some rows it is Zero even though the date is in the past.

Assume today's data is 22 March 2025 and Date of Birth given is 31 December 2024

Excepted Age will be Zero that is person not yet 1 year old. The code calculates the completed years between today and date of birth. It will be zero as it is not yet one year.

We can modify the code to calculate the age in months or days.

I will stop it over here.

30. Open the crontab editor and paste the command and the process is automated.

```
  GNU nano 7.2                                                    /
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 0 * * * /usr/bin/php /var/www/html/update_age.php
```

31. Attached php code


index.php


update_age.php