**PASSPORT AUTOMATION SYSTEM**

**DATE:**

**AIM :**

To develop the Passport Automation System using rational rose tools, visual basic and MS access

**PROBLEM ANALYSIS AND PROJECT PLAN**

To simplify the process of applying passport, software has been created by designing through rational rose tool, using visual basic as a front end and Microsoft access as a back end. Initially the applicant login the passport automation system and submits his details. These details are stored in the database and verification process done by the passport administrator, regional administrator and police the passport is issued to the applicant.

**PROBLEM STATEMENT**

• Passport Automation System is used in the effective dispatch of passport to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner.

• The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.

• This forms the first and foremost step in the processing of passport application. After the first round of verification done by the system, the information is in turn forwarded to the regional administrator's (Ministry of External Affairs) office.

• The application is then processed manually based on the report given by the system, and any forfeiting identified can make the applicant liable to penalty as per the law.

• The system forwards the necessary details to the police for its separate verification whose report is then presented to the administrator. After all the necessary criteria have been met, the original information is added to the database and the passport is sent to the applicant.

**INTRODUCTION**

Passport Automation System is an interface between the Applicant and the Authority responsible for the Issue of Passport. It aims at improving the efficiency in the Issue of Passport and reduces the complexities involved in it to the maximum possible extent.

**PURPOSE**

If the entire process of 'Issue of Passport' is done in a manual manner then it would take several months for the passport to reach the applicant. Considering the fact that the number of applicants for passport is increasing every year, an Automated System become essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it

**SCOPE**

• The System provides an online interface to the user where they can fill in their personal details

• The authority concerned with the issue of passport can use this system to reduce his workload and process the application in a speedy manner.

• Provide a communication platform between the applicant and the administrator.

• Transfer of data between the Passport Issuing Authority and the Local Police for verification of applicant's information

**SYSTEM FUNCTIONS**

• Secure Registration of information by the Applicants.

• Message box for Passport Application Status Display by the Administrator.

• Administrator can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

**USER CHARACTERISTICS**

• **Applicant** - They are the people who desires to obtain the passport and submit the information to the database.

• **Administrator** - He has the certain privileges to add the passport status and to approve the issue of passport. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of passport.

• **Police** - He is the person who upon receiving intimation from the PAS, perform a personal verification of the applicant and see if he has any criminal case against him before or at

**Introduction to star uml**

StarUML is a sophisticated software modeler aimed to support agile and concise modeling.

The main targets of users are:

* Agile and small development teams
* Professional persons
* Educational institutes

The key features of StarUML are:

* Multi-platform support (MacOS, Windows and Linux)
* UML 2.x standard compliant
* SysML support
* Entity-Relationship diagram (ERD)
* Data-flow diagram (DFD)
* Flowchart diagram
* Multiple windows
* Modern UX
* Dark and light themes
* Retina (High-DPI) display support
* MacPro Pro's Touch Bar support
* Model-driven development
* Open APIs
* Various third-party extensions
* Asynchronous model validation
* Export to HTML docs
* Automatic updates.

Definitions:

Administrator - Refers to the super user who is the Central Authority who has been

vested with the privilege to manage the entire system. It can be any higher official in

the Regional Passport Office of Ministry of External Affairs.

• Applicant - One who wishes to obtain the Passport.

• PAS - Refers to this Passport Automation System.

• HTML - Markup Language used for creating web pages.

• HTTP - Hyper Text Transfer Protocol.

• TCP/IP – Transmission Control Protocol/Internet Protocol is the communication

**CONSTRAINTS**

• The applicants require a computer to submit their information.

• Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.

• The user has to be careful while submitting the information. Much care is required

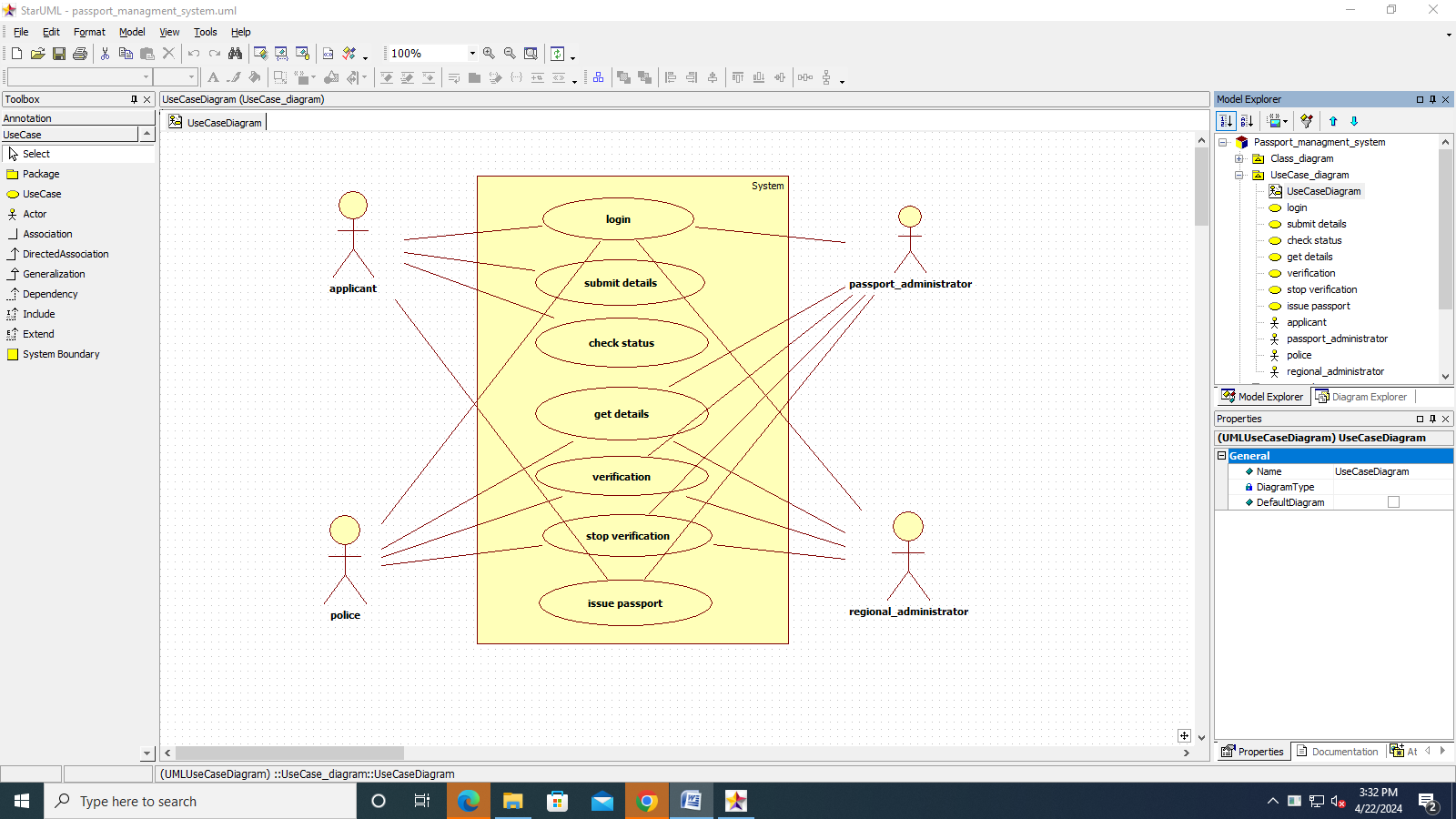
**ASSUMPTIONS AND DEPENDENCIES**

• The Applicants and Administrator must have basic knowledge of computers and English Language.

• The applicants may be required to scan the documents and send

**USE CASE DIAGRAM**

Use case is shown as an ellipse containing the name of use case .An actor is shown as a stick figure with the name below it. Use case diagram is a graph of actors.



**DOCUMENTATION OF USECASE DIAGRAM**

• The actors in use case diagram are Applicant, regional administrator, database, passport Administrator, Police.

• The use cases are Login, givedetails, logout, collectdetails, verification, issue.

• The actors use the use case are denoted by the arrow

• The login use case checks the username and password for applicant, regional administrator, passport administrator and police.

• The submit details use case is used by the applicant for submitting his details

• The check status use case is used by the applicant for checking the status of the application process.

• The get details, verify and store verification use case is used by passport administrator, regional administrator, and police.

• The details use case is used for getting the details form the database for verification

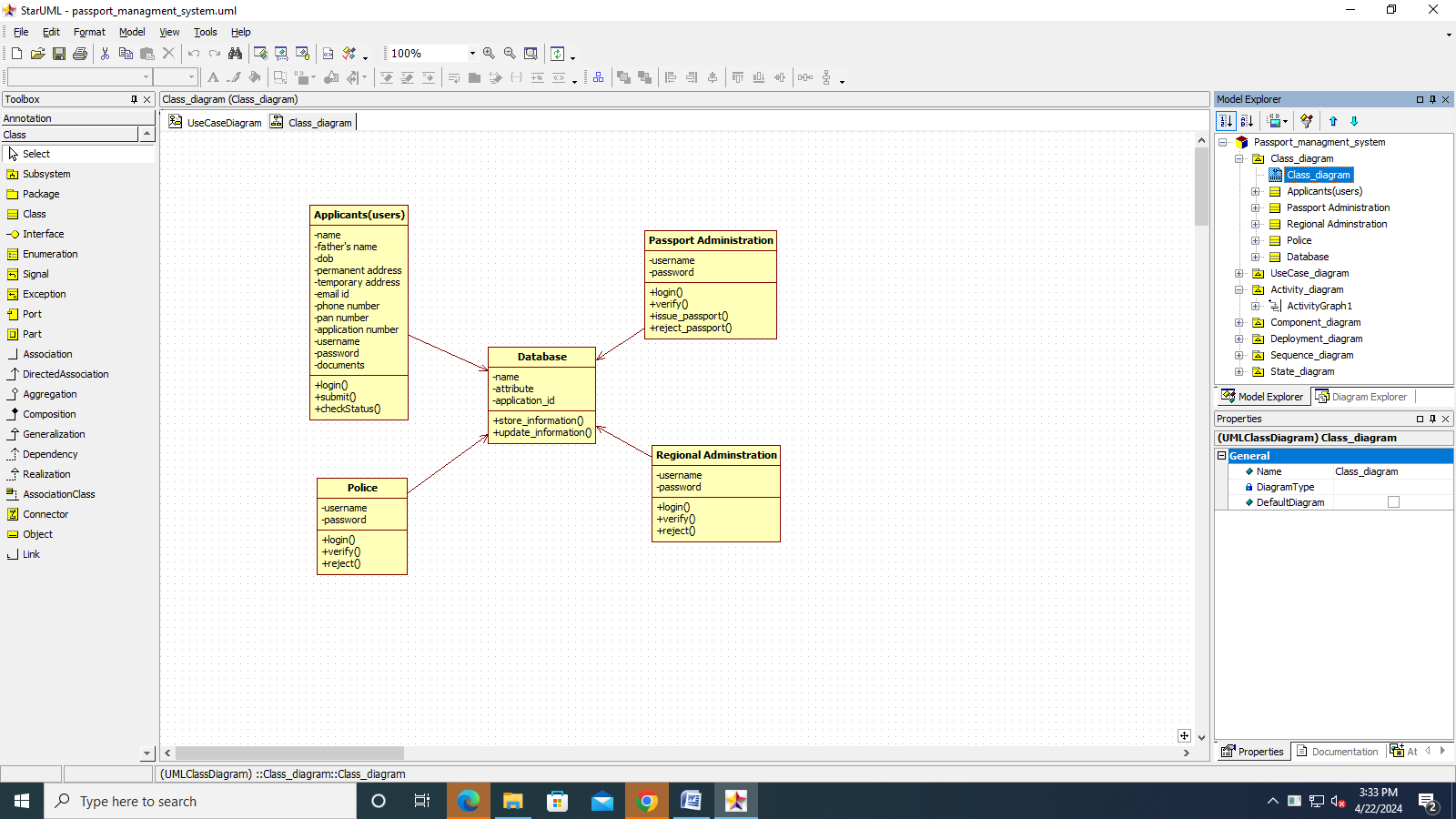
• The verify use case is used for verifying the details by comparing the data in the database.

• The store verification use case is to update the data in the database

• And finally the issue passport use case is used by the passport administrator for issuing passport who’s application verified successfully by all the actor

**CLASSDIAGRAM**

A class is drawn as rectangle box with three compartments or components separated by horizontal lines. The top compartment holds the class name and middle compartment holds the attribute and bottom compartment holds list of operations.



**DOCUMENTATION OF CLASS DIAGRAM**

a. **APPLICANT**-The applicant has attribute such as name and password and operations are login, givedetails and logout. The applicant login and fill the details that are required for applying the passport .After applying the person can view the status of the passport verification process

b. **THE DATABASE**-The database has attributed such as name and operation is store. The purpose is to store the data.

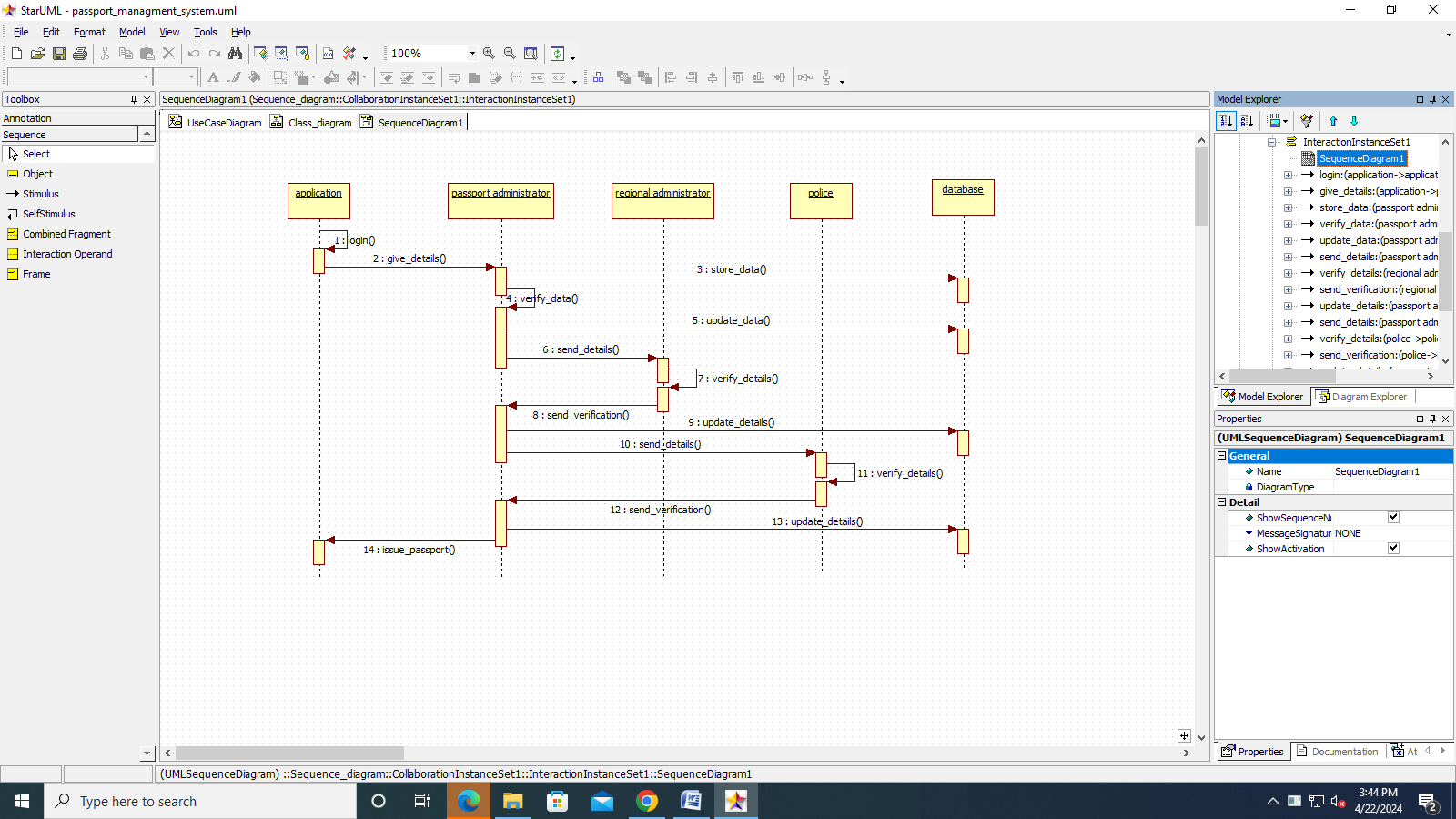
c. **REGIONAL ADMINISTRATOR**- The regional administrator has attribute such as name and operation are get details, verify details and send. The regional administrator get the details form database and verify with their database

d. **PASSPORT** **ADMINISTRATOR**-The passport administrator has attributed such as name and operation are get details, verify details and issue. The passport administrator get the details form database and verify with their database , update the verification and issue the passport

e. **THE** **POLICE**-The police has attribute such as name and operation are get details, verify details and send. The police get the details form database and verify with their database , update the verification in the database

**SEQUENCE** **DIAGRAM**

A sequence diagram shows an interaction arranged in time sequence, It shows object participating in interaction by their lifeline by the message they exchange arranged in time sequence. Vertical dimension represent time and horizontal dimension represent object.



**DOCUMENTATION OF SEQUENCE DIAGRAM.**

a. **The** **applicant** login the database and give his details and database store the details.

b. **The** **passport** **administrator** get the details from the database and do verification and the forward to regional administrator.

c. **The** **regional** **administrator** get details form passport administrator and perform verification and send report to passport administrator.

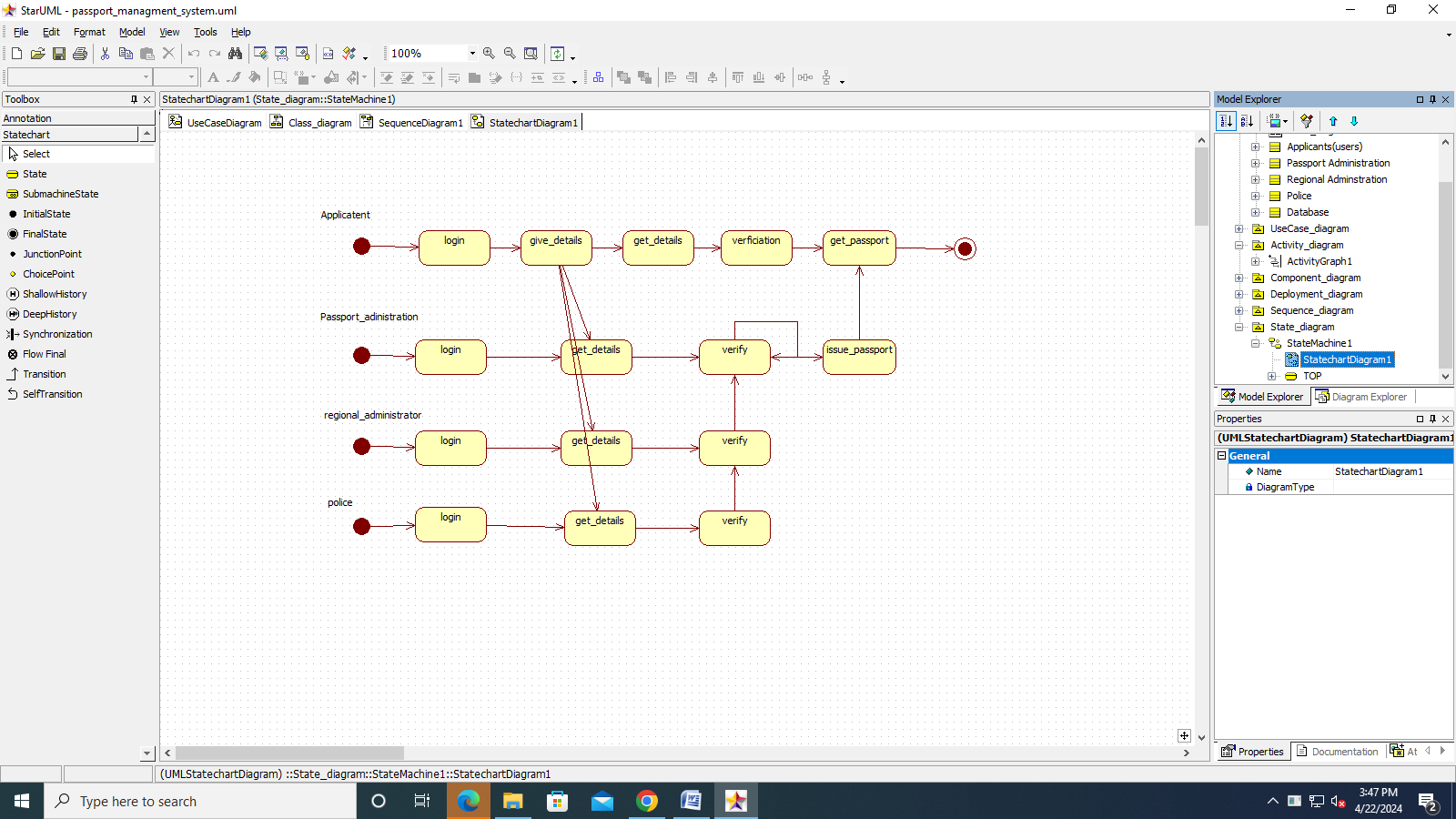
d. **The** **police** get the details form passport administrator and perform verification and send report to passport administrator

**COLLABORATION** **DIAGRAM**

A collaboration diagram is similar to sequence diagram but the message in number format. In a collaboration diagram sequence diagram is indicated by the numbering the message. A collaboration diagram, also called a communication diagram or interaction diagram, A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time

**STATE** **CHART** **DIAGRAM**

The state chart diagram contains the states in the rectangle boxes and starts in indicated by the dot and finish is indicated by dot encircled. The purpose of state chart diagram is to understand the algorithm in the performing method.



**DOCUMENTATION** **OF** **STATE** **CHART** **DIAGRAM**

a. The states of the passport automation system are denoted in the state chart diagram

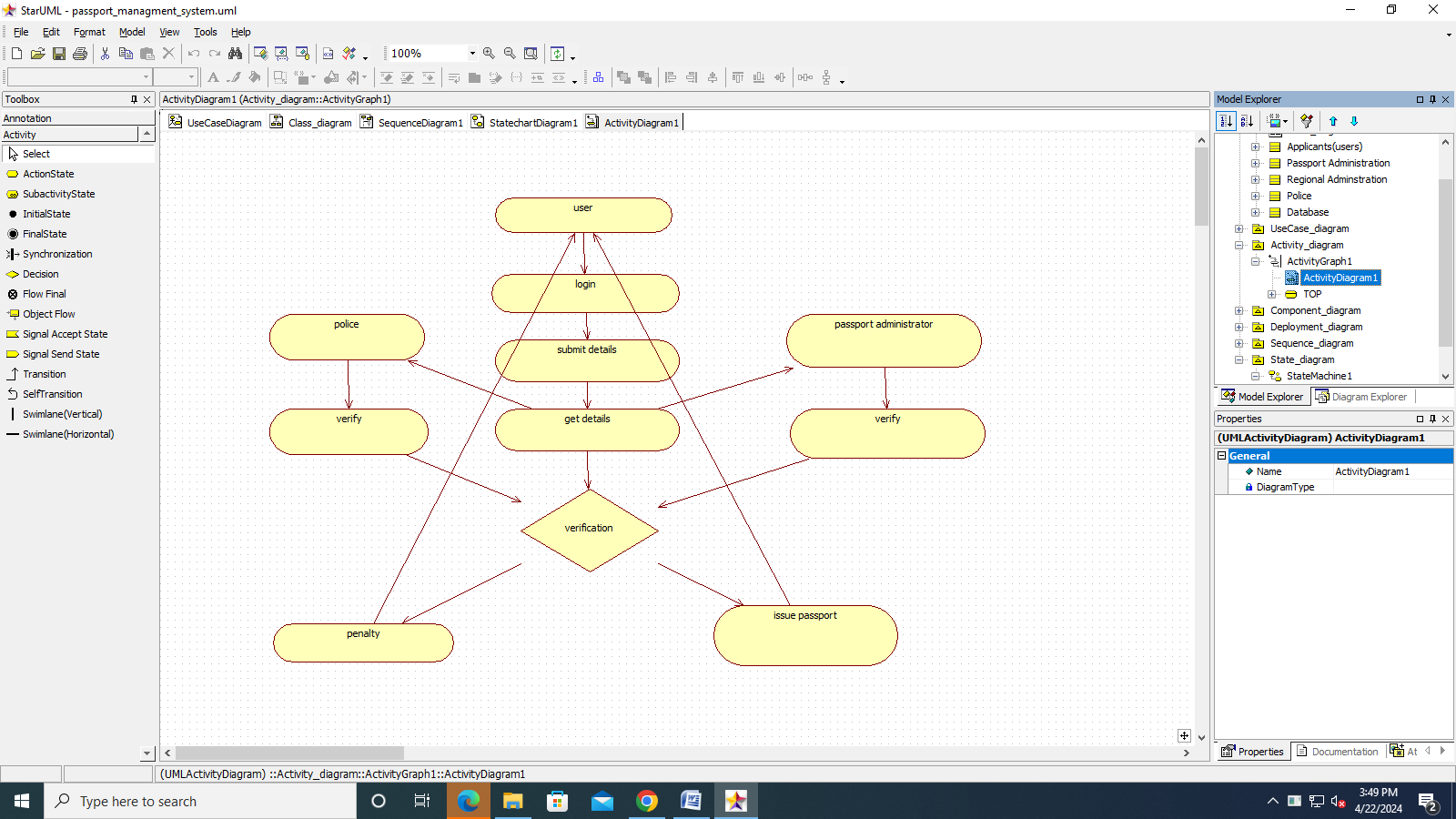
b. Login state represent authentication for login the passport automation system.

c. In this state, it checks whether the applicant has provided all the details that is required.

d. Police, regional administrator and passport administrator get necessary details and verification of the applicant are denoted from the Get detail state and verification state

**ACTIVITY** **DIAGRAM**

An activity diagram is a variation or special case of a state machine in which the states or activity representing the performance of operation and transitions are triggered by the completion of operation. The purpose is to provide view of close and what is going on inside a use case or among several classes. An activity is shown as rounded box containing the name of operation

****

**DOCUMENTATION** **OF** **ACTIVITY** **DIAGRAM**

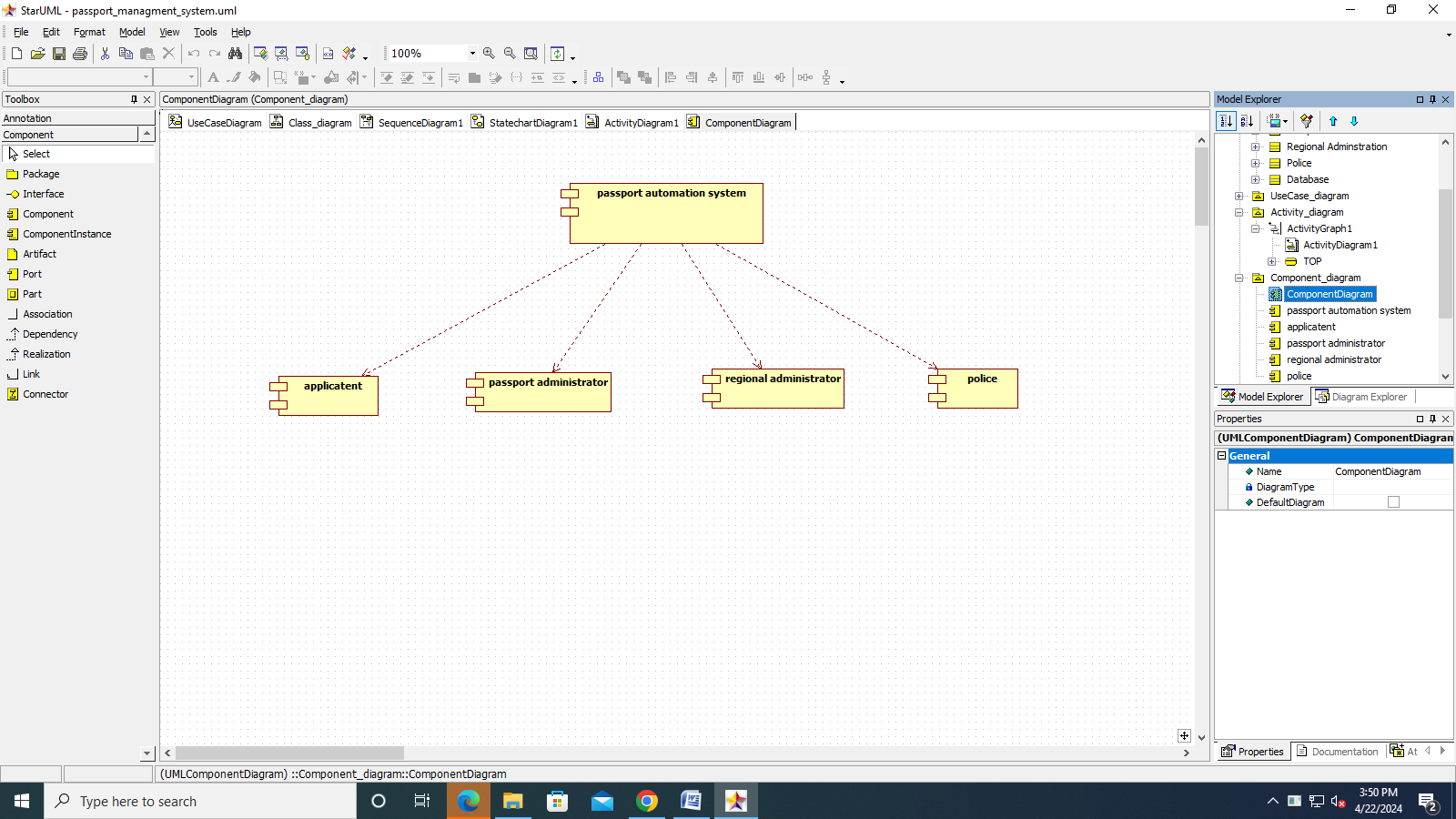
a. The activities in the passport automation system are login, submit details, get details, issue passport and penalty and verification.

b. In the login activity applicant give username and password and then login into the passport automation system after then fill the details that are required for application.

c. After the verification procedure completed successfully the passport is issued to the applicant.

**COMPONENT** **DIAGRAM**

The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication assosiation.

****

**DOCUMENTATION** **OF** **COMPONENT** **DIAGRAM**

a. The components in the passport automation system are passport automation system, applicant, passport administrator, regional administrator, and police.

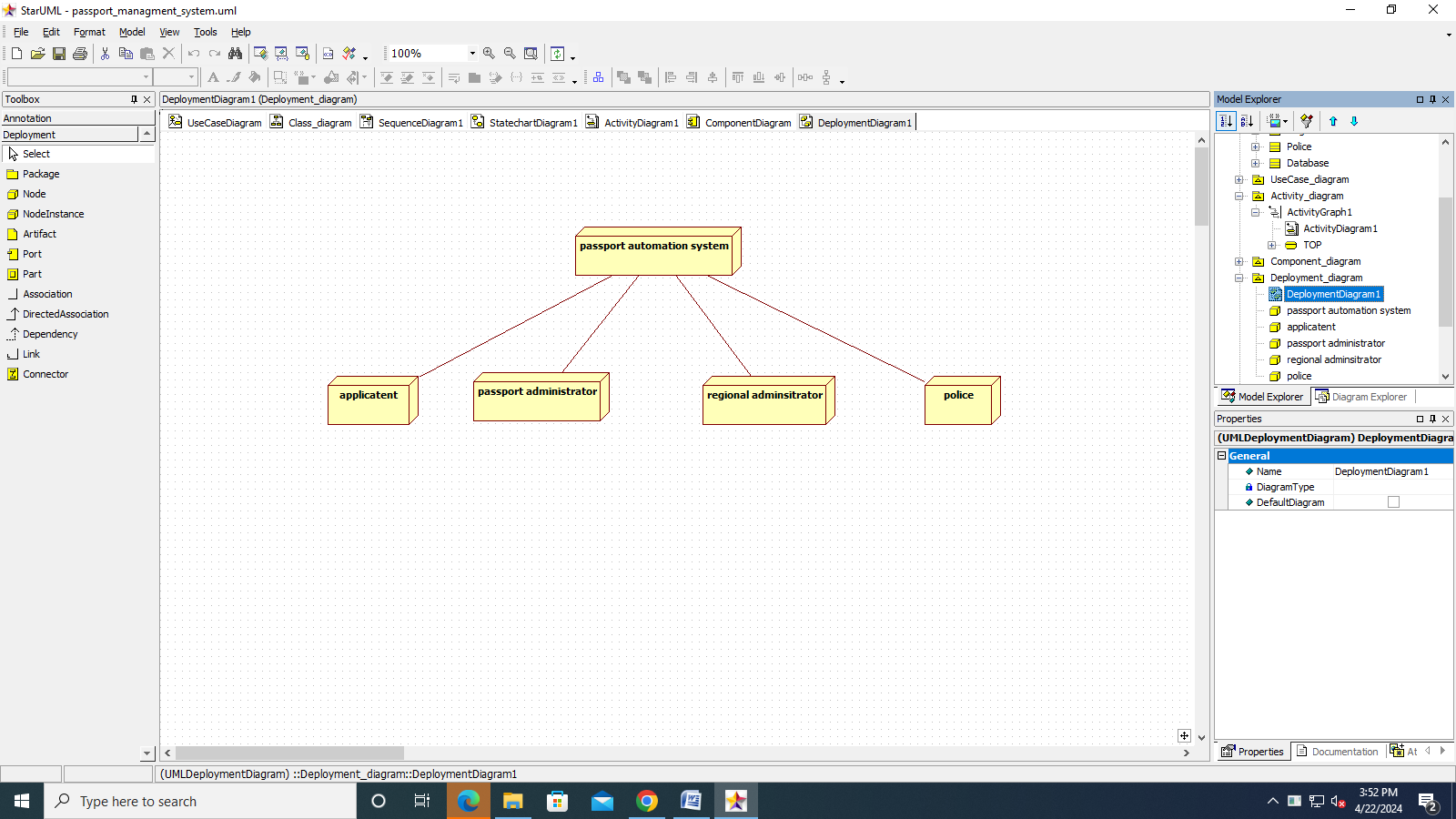
b. Applicant ,passport administrator, regional administrator and police are dependent on passport automation system are shown by the dotted arrow

**DEPLOYMENT** **DIAGRAM**

It is a graph of nodes connected by communication association. It is represented by a three dimensional box. A deployment diagram in the unified modeling language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimentional box. Dependencies are represented by communication association. The basic element of a deployment diagram is a node of two types

**DEVICE** **NODE**– A physical computing resource with processing and memory service to execute software, such as a typical computer or a mobile phone.

**EXECUTION** **ENVIRONMENT** **NODE** This is a software computing resource that runs within an outer node and which itself provides a service to host an execute other executable software element.

****

**DOCUMENTATION** **OF** **DEPLOYMENT** **DIAGRAM**

The device node is passport automation system and execution environment node are applicant passport administrator, regional administrator, and police.

**PACKAGE** **DIAGRAM**

A package diagram is represented as a folder shown as a large rectangle with a top attached to its upper left corner. A package may contain both sub ordinate package and ordinary model elements. All uml models and diagrams are organized into package. A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).

There are three types of layer. They are

o User interface layer

o Domain layer

o Technical services layer

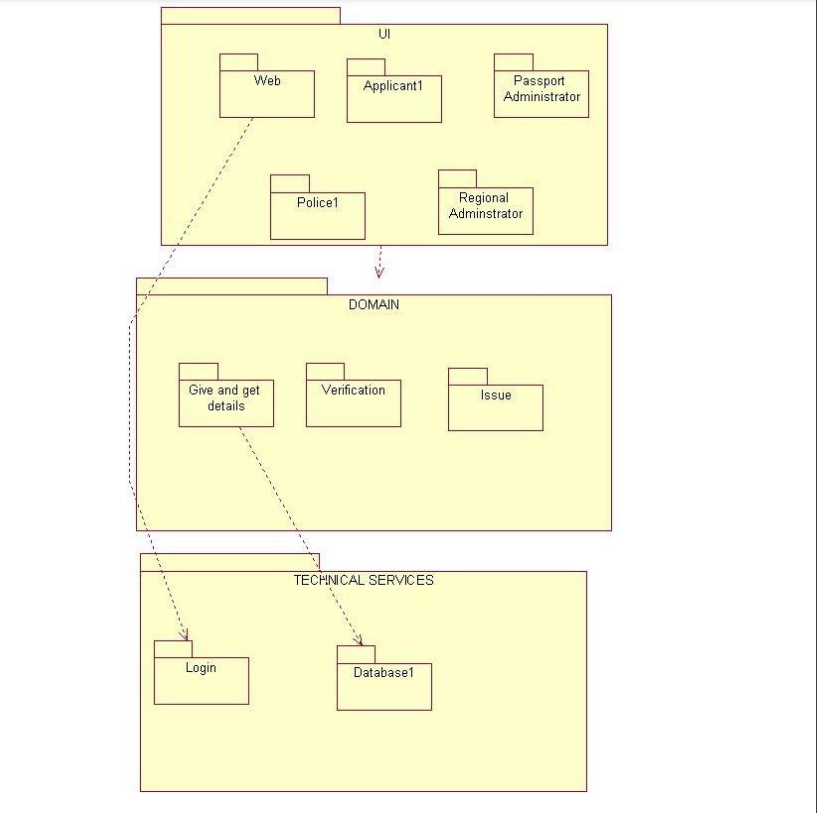
**DOCUMENTATION** **OF** **PACKAGE** **DIAGRAM**

The three layer in the passport automation system are user interface layer, domain layer, technical service layer

a. The user interface layer- represents the user interface components such as web, applicant, passport administrator, police, and regional administrator.

b. The domain layer- has major actions such as give and get details, verification and issues.

c. Technical service layer- authenticated user only can access the technical services.

****

**SOURCE CODE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>apply passport</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="apply\_passport\_boby">

<form action="apply\_passport" method="POST">

<h1>apply passport</h1>

<div id="apply\_passport\_content">

<div id="apply\_passport\_main">

<div id="apply\_passport\_profile">

<img src="{{url\_for('static',filename='images/profile\_image\_default.png')}}"/>

</div>

<div id="apply\_passport\_application\_id">

<label>application id:</label>

<label>auto generated</label>

</div>

<div id="apply\_passport\_name">

<label>name:</label>

<input type="text" name="apply\_passport\_name" />

</div>

<div id="apply\_passport\_father\_name">

<label>father name:</label>

<input type="text" name="apply\_passport\_father\_name"/>

</div>

<div id="apply\_passport\_mother\_name">

<label>mother name:</label>

<input type="text" name="apply\_passport\_mother\_name"/>

</div>

<div id="apply\_passport\_dob">

<label>dob:</label>

<input type="date" name="apply\_passport\_dob"/>

</div>

<div id="apply\_passport\_phone">

<label>phone:</label>

<input type="text" name="apply\_passport\_phone"/>

</div>

<div id="apply\_passport\_adderss">

<label>address:</label>

<input type="text" name="apply\_passport\_adderss"/>

</div>

<div id="apply\_passport\_email">

<label>email:</label>

<input type="text" name="apply\_passport\_email"/>

</div>

<div id="apply\_passport\_document">

<label id="apply\_passport\_document\_heading">document</label>

</div>

<div id="apply\_passport\_id\_proof">

<select id="apply\_passport\_id\_proof\_selector" name="apply\_passport\_id\_proof">

<option value="aadhar\_card">aadhar card</option>

<option value="ration\_card">ration card</option>

<option value="licence">licence</option>

<option value="voter\_id">voter id</option>

</select>

<input type="file" value="upload" name="apply\_passport\_id\_proof\_file"/>

</div>

<div id="apply\_passport\_10th\_mark\_sheet">

<label>10th mark sheet:</label>

<input type="file" value="upload" name="apply\_passport\_10th\_mark\_sheet"/>

</div>

<div id="apply\_passport\_12th\_mark\_sheet">

<label>12th mark sheet:</label>

<input type="file" value="upload" name="apply\_passport\_12th\_mark\_sheet"/>

</div>

<div id="apply\_passport\_pan\_card">

<label>pan card:</label>

<input type="file" value="upload" name="apply\_passport\_pan\_card"/>

</div>

</div>

</div>

<div id="apply\_passport\_button">

<input type="submit" id="apply\_passport\_button\_button" value="submit"/>

</div>

</form>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>change password</title>

<!-- <link rel="stylesheet" href="style.css"> -->

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="change\_password\_body">

<div id="change\_password\_content">

<div id="change\_password\_main">

<h1>change password</h1>

<div id="change\_password\_password">

<label>password:</label>

<input type="text"/>

</div>

<div id="change\_password\_newpassword">

<label>new password:</label>

<input type="password"/>

</div>

<div id="change\_password\_reenter">

<label>reenter:</label>

<input type="text"/>

</div>

<div id="change\_password\_button">

<input type="button" value="change password">

</div>

</div>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>check status</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="check\_status\_boby">

<h1>check status</h1>

<div id="check\_status\_content">

<div id="check\_status\_main">

<div id="check\_status\_application\_number">

<label>enter application number:</label>

<input type="text"/>

</div>

<div id="check\_status\_aadhar\_number">

<label>aadhar numder:</label>

<input type="text"/>

</div>

<div id="check\_status">

<input type="button" value="check status"/>

</div>

</div>

</div>

</div>

</body>

</html><!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>check status</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="check\_status\_boby">

<h1>check status</h1>

<div id="check\_status\_content">

<div id="check\_status\_main">

<div id="check\_status\_application\_number">

<label>enter application number:</label>

<input type="text"/>

</div>

<div id="check\_status\_aadhar\_number">

<label>aadhar numder:</label>

<input type="text"/>

</div>

<div id="check\_status">

<input type="button" value="check status"/>

</div>

</div>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>forget password</title>

<!-- <link rel="stylesheet" href="style.css"> -->

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="forget\_password\_nav">

<a href="{{url\_for('home')}}">home</a>

<a href="{{url\_for('user\_login')}}" id="forget\_password\_login\_link">login</a>

</div>

<div id="forget\_password\_body">

<div id="forget\_password\_content">

<div id="forget\_password\_main">

<h1>forget password</h1>

<div id="forget\_password\_username">

<label>username:</label>

<input type="text"/>

</div>

<div id="forget\_password\_aadharno">

<label>aadherNo:</label>

<input type="text"/>

</div>

<div id="forget\_password\_password">

<label>password:</label>

<input type="password"/>

</div>

<div id="forget\_password\_reenter">

<label>re enter:</label>

<input type="text"/>

</div>

<div id="forget\_password\_changepassword">

<input type="button" value="change password"/>

</div>

</div>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>home</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="home\_body">

<div id="home\_content">

<div id="home\_menu">

<div id="home\_login\_menu" >

<a href="{{url\_for('user\_login')}}">login</a>

<!-- <option value="register">register</option> -->

</div>

<a href="{{url\_for('check\_status')}}">status</a>

<img id="home\_profile\_image" src="{{url\_for('static', filename='images/profile\_image\_default.png')}}"/>

</div>

<div id="home\_image">

<img src="{{url\_for('static', filename='images/passport-seva-inner14-new.jpeg')}}"/>

</div>

<div id="home\_main">

<div id="home\_application">

<h1>passport application</h1>

<p>

requirements:<br>

-name<br>

-phone(300kb)<br>

-id proof(2mb)<br>

-address<br>

-phone<br>

-email id<br>

-pan card<br>

-mark sheet(10th,12th)(3mb)<br>

-voter id(3mb)<br>

-dob<br>

-father name<br>

-mother name<br>

</p>

</div>

<div id="home\_how\_to\_apply">

<h1>home how to apply</h1>

<p>

<a href="{{url\_for('apply\_passport')}}">1.apply for passport</a> <br>

<a href="{{url\_for('make\_payment')}}">2.make payment</a><br>

<a href="{{url\_for('check\_status')}}">3.police verification</a> <br>

<a href="{{url\_for('check\_status')}}">4.get passport</a> <br>

</p>

</div>

</div>

</div>

</div>

<script>

document.getElementById("home\_profile\_image").addEventListener("click", function() {

// Redirect to the desired URL

window.location.href = "{{ url\_for('user\_profile') }}";

});

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Make payment</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="make\_payment\_body">

<h1>make payment</h1>

<div id="make\_payment\_content">

<div id="make\_payment\_main">

<div id="make\_payment\_application\_id">

<label>application id:</label>

<lable name="make\_payment\_application\_id">auto generated id</lable>

</div>

<div id="make\_payment\_application\_status">

<label>applicathion status: </label>

<label>payment pending</label>

</div>

<div id="make\_payment\_pay">

<input type="submit" value="pay now"/>

<input type="button" id="make\_payment\_pay\_later" value="pay later" />

</div>

</div>

</div>

</div>

<script>

document.getElementById("make\_payment\_pay\_later").addEventListener("click", function() {

// Redirect to the desired URL

window.location.href = "{{ url\_for('home') }}";

});

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>new user register</title>

<!-- <link rel="stylesheet" href="style.css"> -->

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="new\_user\_register\_body">

<h1>new user register</h1>

<div id="new\_user\_register\_content">

<form action="register\_user" method="POST">

<div id="new\_user\_register\_main">

<div id="new\_user\_register\_profile\_image">

<img src="{{url\_for('static',filename='images/profile\_image\_default.png')}}">

</div>

<div id="new\_user\_register\_name">

<label>name:</label>

<input type="text" name="new\_user\_register\_name"/>

</div>

<div id="new\_user\_register\_aadharnumber">

<label>aadherNo:</label>

<input type="text" name="new\_user\_register\_aadharnumber"/>

</div>

<div id="new\_user\_register\_phone">

<label>phone:</label>

<input type="text" name="new\_user\_register\_phone"/>

</div>

<div id="new\_user\_register\_email">

<label>email:</label>

<input type="text" name="new\_user\_register\_email"/>

</div>

<div id="new\_user\_register\_address">

<label>address:</label>

<input type="text" name="new\_user\_register\_address"/>

</div>

<div id="new\_user\_register\_fathersname">

<label>father name:</label>

<input type="text" name="new\_user\_register\_fathersname"/>

</div>

<div id="new\_user\_register\_mothername">

<label>mother name:</label>

<input type="text" name="new\_user\_register\_mothername"/>

</div>

<div id="new\_user\_register\_dob">

<label>dob:</label>

<input type="date" name="new\_user\_register\_dob"/>

</div>

<div id="new\_user\_register\_username">

<label>user name:</label>

<input type="text" name="new\_user\_register\_username"/>

</div>

<div id="new\_user\_register\_password">

<lable>password:</lable>

<input type="password" name="new\_user\_register\_password"/>

</div>

<div id="new\_user\_register\_reenter">

<label>reenter:</label>

<input type="text" name="new\_user\_register\_reenter"/>

</div>

<div id="new\_user\_register\_login">

<a href="{{url\_for('user\_login')}}">login</a>

<input type="submit" value="register" name="new\_user\_register\_login"/>

</div>

</div>

</form>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>status result</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="status\_result\_boby">

<div id="status\_result\_content">

<div id="status\_result\_main">

<div id="status\_result\_application\_id">

<label>application id:</label>

<label>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_</label>

</div>

<div id="status\_result\_status">

<label>status:</label>

<label>passport generated</label>

</div>

<div id="status\_result\_button">

<a href="{{url\_for('home')}}">back home</a>

<input type="button" value="get passport" style="display:none;"/>

</div>

</div>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>user login</title>

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

<!-- <link rel="stylesheet" href="style.css"/> -->

</head>

<body>

<div id="user\_login\_body">

<form action="/login" method="post">

<div id="user\_login\_content">

<div id="user\_login\_main">

<div id="user\_login\_heading">

<h1>user login</h1>

</div>

<div id="user\_login\_username">

<label>username:</label>

<input type="text" name="user\_login\_username"/>

</div>

<div id="user\_login\_password">

<label>password:</label>

<input type="password" name="user\_login\_password"/>

</div>

<div id="user\_login\_captcha">

<label>captcha:</label>

<input type="text"/>

</div>

<div id="user\_login\_captcha\_number">

<label>1234</label>

</div>

<div id="user\_login\_loginbutton">

<input id="user\_login\_button" type="submit" value="login">

</div>

<div id="user\_login\_register">

<a href="{{url\_for('show\_register\_user\_form')}}">register</a>

<a href="{{url\_for('forget\_password')}}">Forget password:</a>

</div>

</div>

</div>

</form>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>user profile</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="{{url\_for('static',filename='style.css')}}">

</head>

<body>

<div id="user\_profile\_boby">

<div id="user\_profile\_content">

<div id="user\_profile\_main">

<h1>user profile</h1>

<div id="user\_profile\_image">

<img id="user\_profile\_image\_image" src="{{url\_for('static', filename='images/profile\_image\_default.png')}}"/>

</div>

<div id="user\_profile\_name">

<label>name:</label>

<input type="text"/>

</div>

<div id="user\_profile\_user\_name">

<label>user name:</label>

<input type="text"/>

</div>

<div id="user\_profile\_father\_name">

<label>father name:</label>

<input type="text"/>

</div>

<div id="user\_profile\_mather\_name">

<label>mother name:</label>

<input type="text"/>

</div>

<div id="user\_profile\_aadhar">

<label>aadhar number:</label>

<input type="text"/>

</div>

<div id="user\_profile\_dob">

<label>dob:</label>

<input type="text"/>

</div>

<div id="user\_profile\_phone">

<label>phone:</label>

<input type="text"/>

</div>

<div id="user\_profile\_email">

<label>email:</label>

<input type="text"/>

</div>

<div id="user\_profile\_button">

<input type="button" value="edit" />

<a href="change\_password\_user\_login.html">change password</a>

</div>

</div>

</div>

</div>

</body>

</html>

**App.py**

from flask import Flask,render\_template,url\_for,request,redirect

from flask\_sqlalchemy import SQLAlchemy

from flask\_login import UserMixin

# from flask\_wtf import wtforms

from wtforms import StringField,PasswordField,SubmitField

from wtforms.validators import InputRequired,Length,ValidationError

from flask\_bcrypt import Bcrypt

app = Flask(\_\_name\_\_)

bcrypt = Bcrypt(app)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///database.db'

app.config['SECRET\_KEY'] = "123"

db = SQLAlchemy(app)

class User(db.Model,UserMixin):

id = db.Column(db.Integer,primary\_key = True)

username = db.Column(db.String(50),nullable=False)

password = db.Column(db.String(80),nullable=False)

@app.route("/")

def home():

return render\_template("home.html")

@app.route("/login", methods=['POST', 'GET'])

def user\_login():

if request.method == 'POST':

uname = request.form['user\_login\_username']

passw = request.form['user\_login\_password']

# print(uname)

passw = bcrypt.generate\_password\_hash(passw)

# return redirect(url\_for("home"))

new\_user = User(username=uname,password=passw)

db.session.add(new\_user)

db.session.commit()

# return "helo world "+str(uname)

return redirect("/")

return render\_template("user\_login.html")

@app.route("/forget\_password")

def forget\_password():

return render\_template("forget\_password\_user\_login.html")

@app.route("/register\_user",methods=['POST', 'GET'])

def show\_register\_user\_form():

if request.method =='POST':

name = request.form['new\_user\_register\_name']

aadharno = request.form['new\_user\_register\_aadharnumber']

phone = request.form['new\_user\_register\_phone']

email = request.form['new\_user\_register\_email']

adderss = request.form['new\_user\_register\_address']

fathername = request.form['new\_user\_register\_fathersname']

mothername = request.form['new\_user\_register\_mothername']

dob = request.form['new\_user\_register\_dob']

username = request.form['new\_user\_register\_username']

password = request.form['new\_user\_register\_password']

reenter = request.form['new\_user\_register\_reenter']

# return "register successful"

return redirect("/")

return render\_template("register\_user\_login.html")

@app.route("/change\_password")

def change\_password():

return render\_template("change\_password\_user\_login.html")

@app.route("/apply\_passport",methods=['POST', 'GET'])

def apply\_passport():

if request.method =='POST':

name = request.form['apply\_passport\_name']

fathername = request.form['apply\_passport\_father\_name']

mothername = request.form['apply\_passport\_mother\_name']

dob = request.form['apply\_passport\_dob']

phone = request.form['apply\_passport\_phone']

adderss = request.form['apply\_passport\_adderss']

email = request.form['apply\_passport\_email']

select = request.form['apply\_passport\_id\_proof']

idprooffile = request.form['apply\_passport\_id\_proof\_file']

mark10th = request.form['apply\_passport\_10th\_mark\_sheet']

mark12th = request.form['apply\_passport\_12th\_mark\_sheet']

pancard = request.form['apply\_passport\_pan\_card']

return redirect("/make\_payment")

return render\_template("apply\_passport.html")

@app.route("/user\_profile")

def user\_profile():

return render\_template("user\_profile.html")

@app.route("/make\_payment")

def make\_payment():

return render\_template("make\_payment.html")

@app.route("/status\_result")

def status\_result():

return render\_template("status\_result.html")

@app.route("/check\_status")

def check\_status():

return render\_template("check\_status.html")

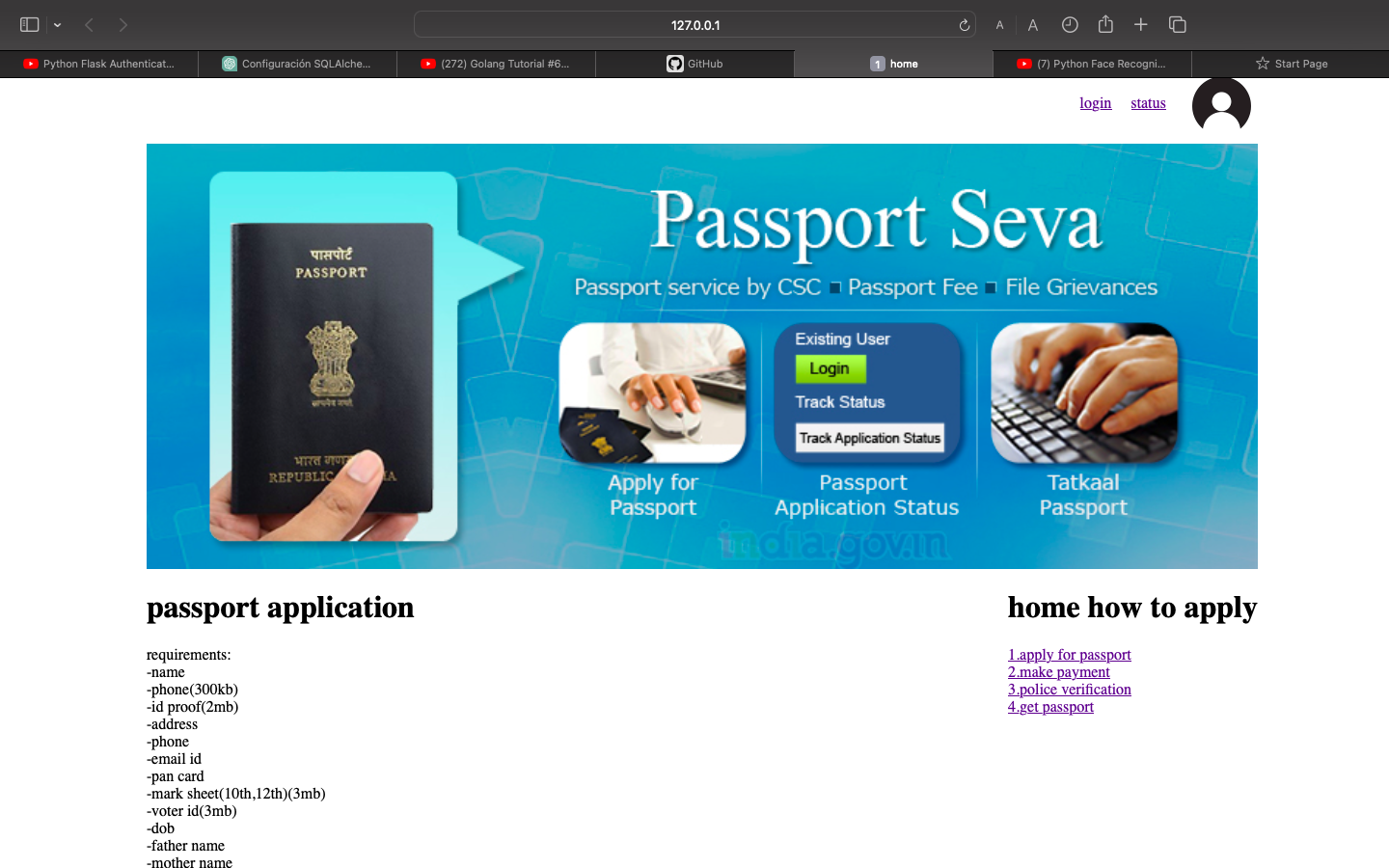
if \_\_name\_\_ == "\_\_main\_\_":

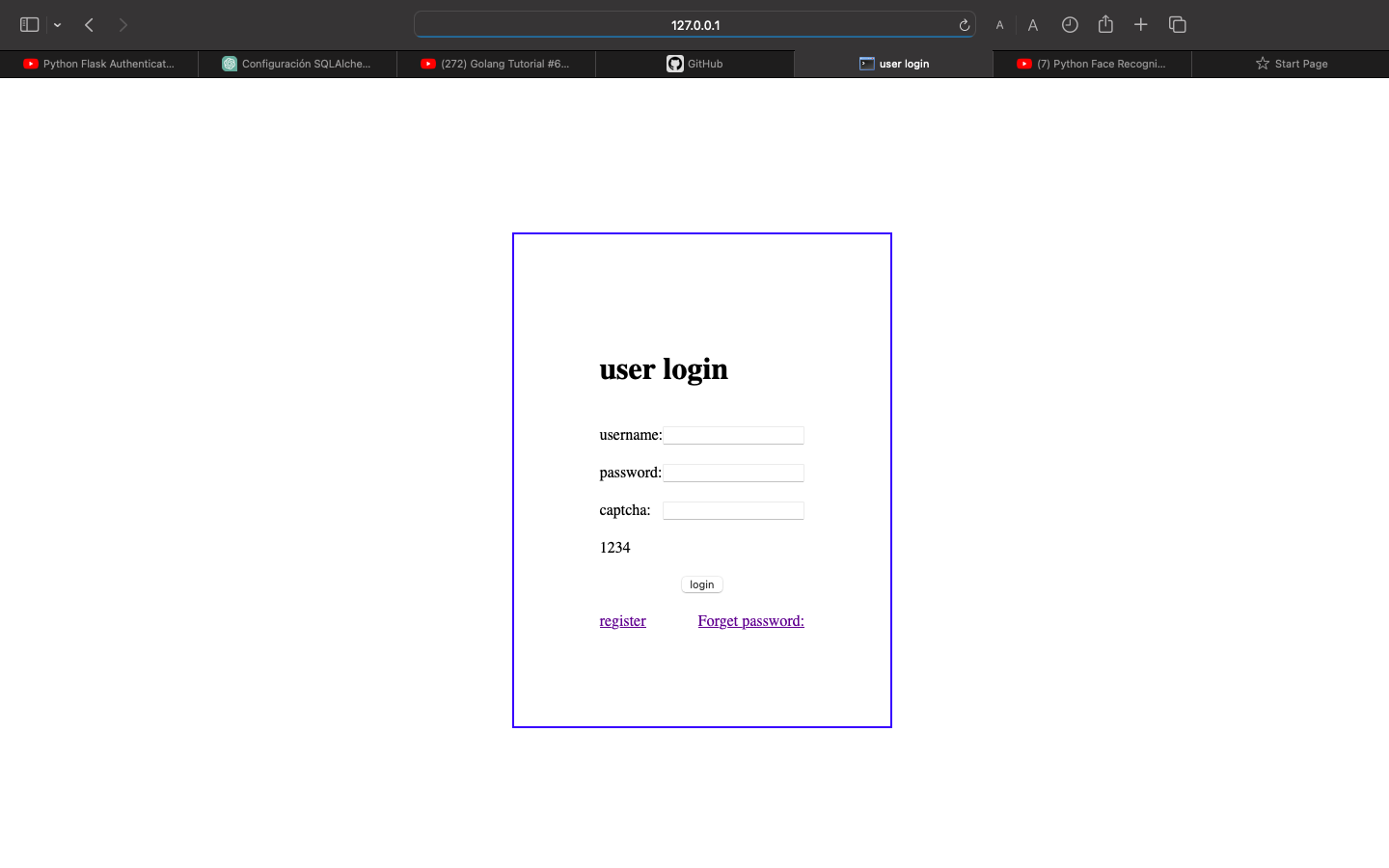
# with app.app\_context():

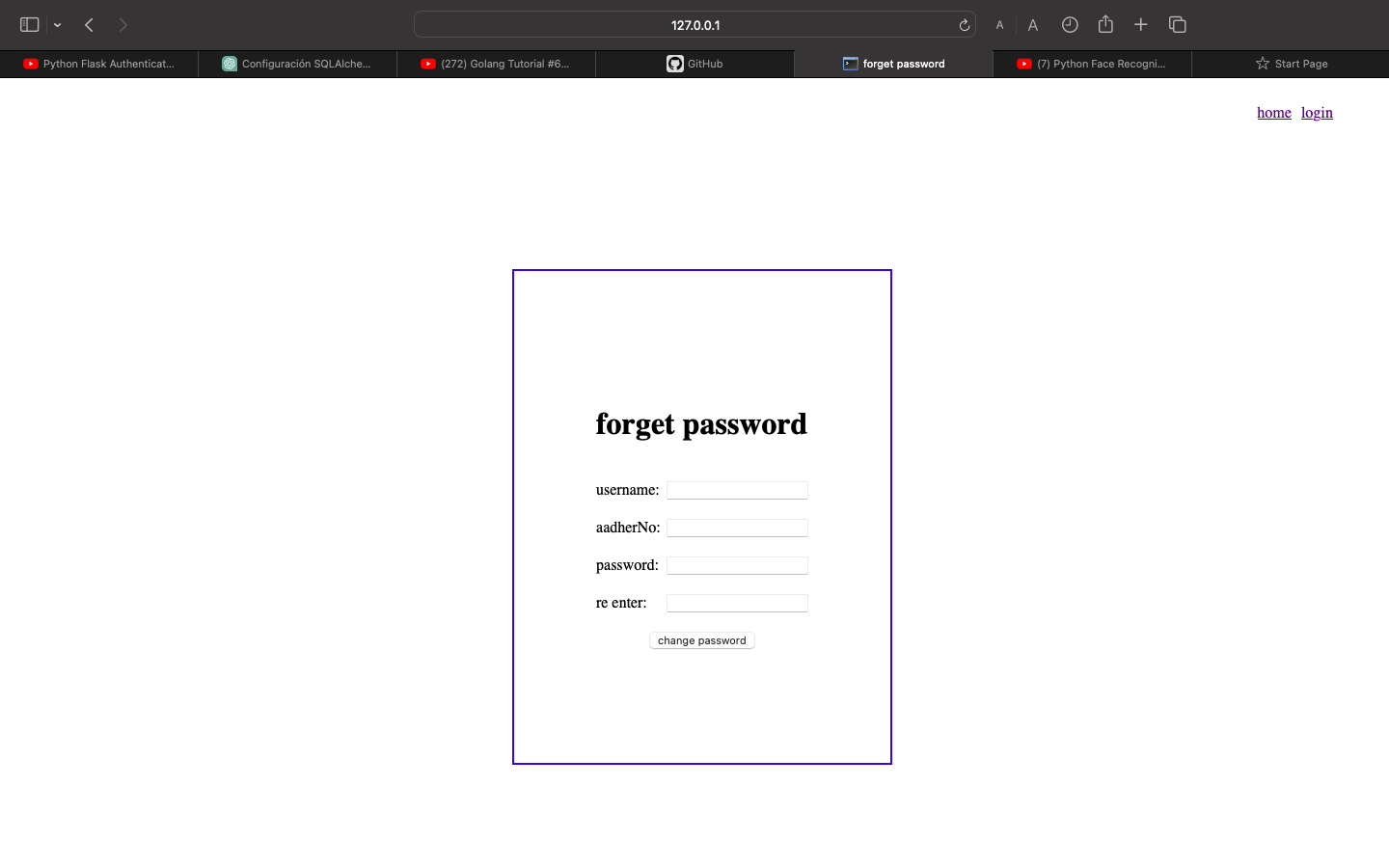
# db.create\_all()

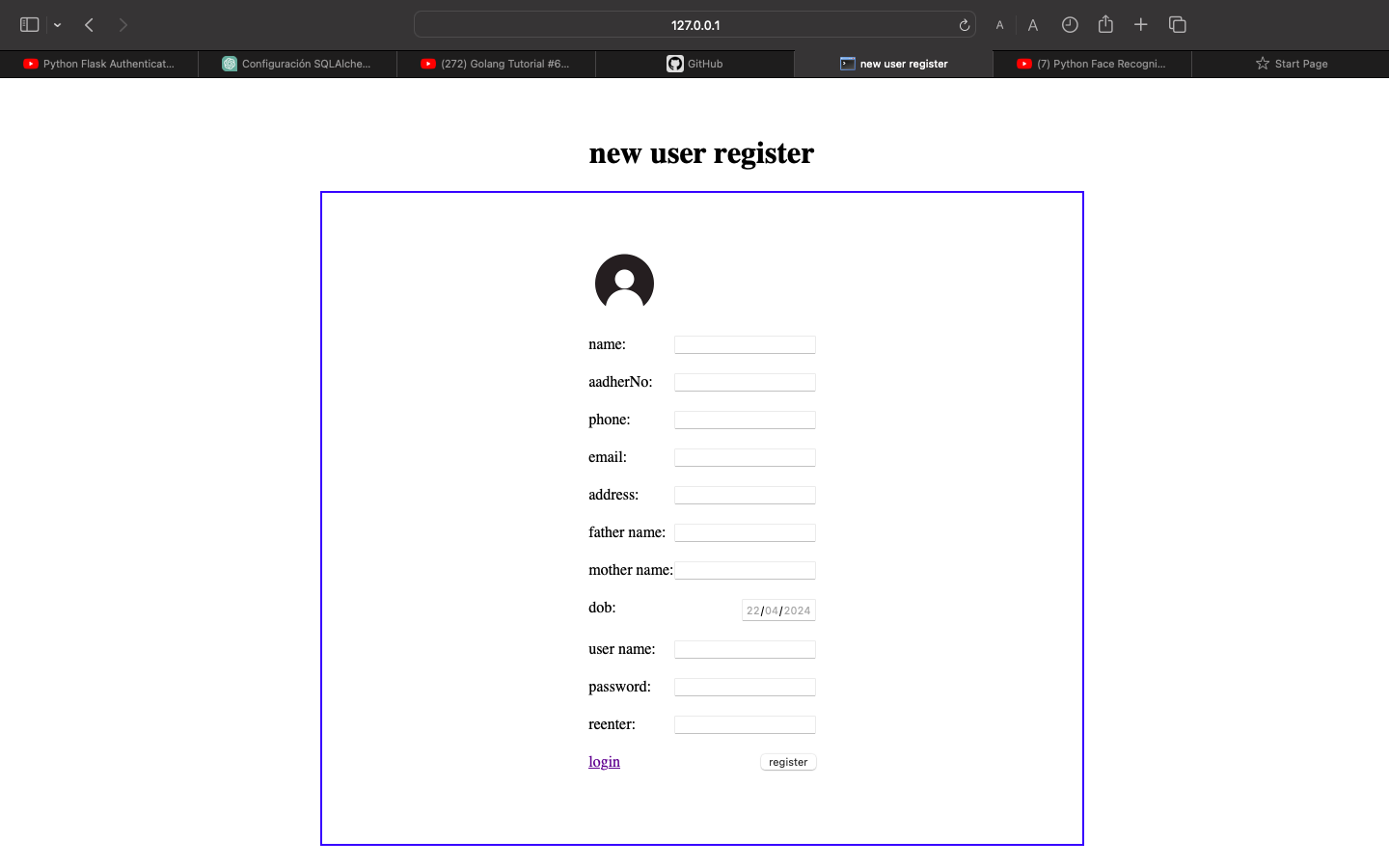
app.run(debug=True)

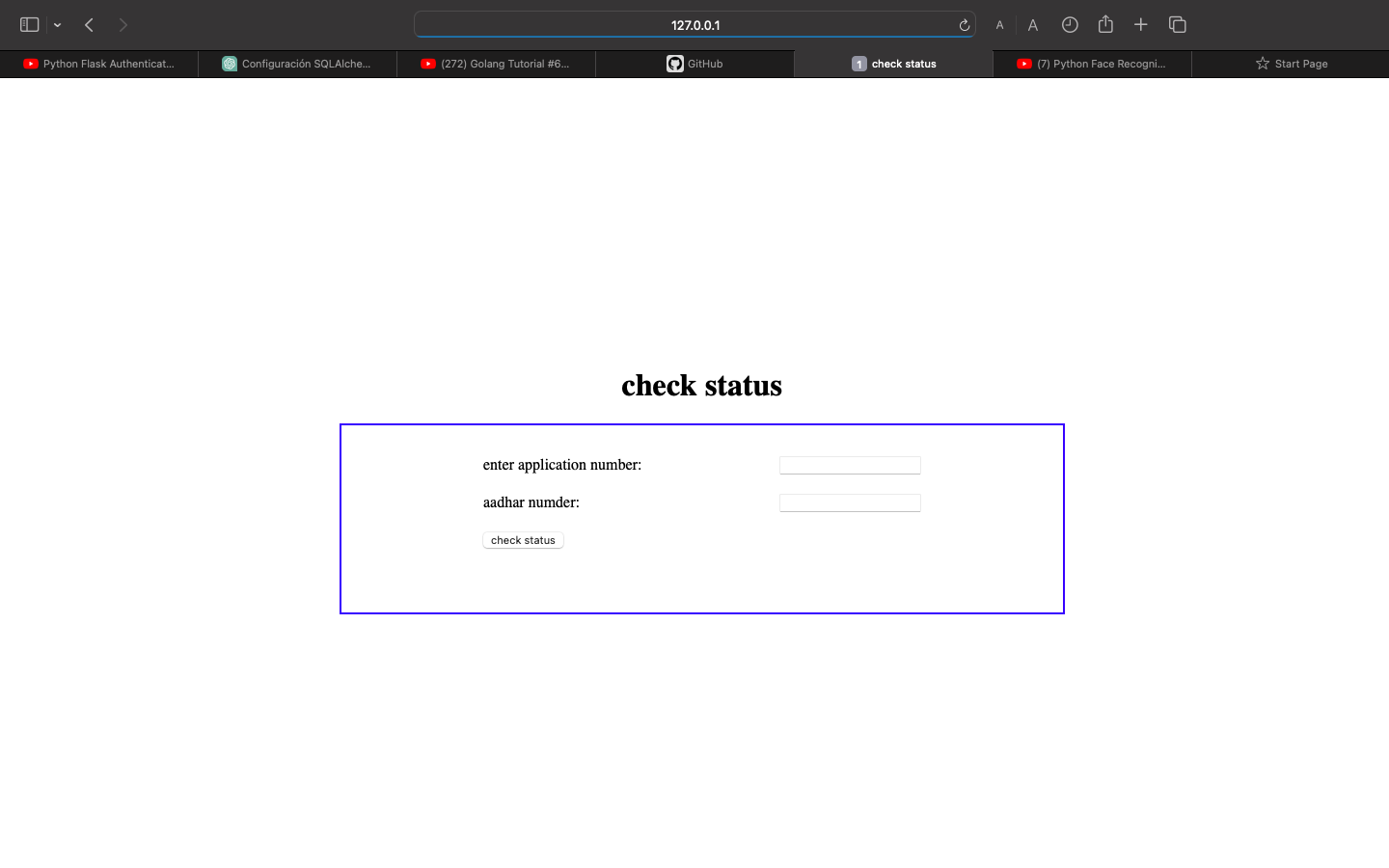
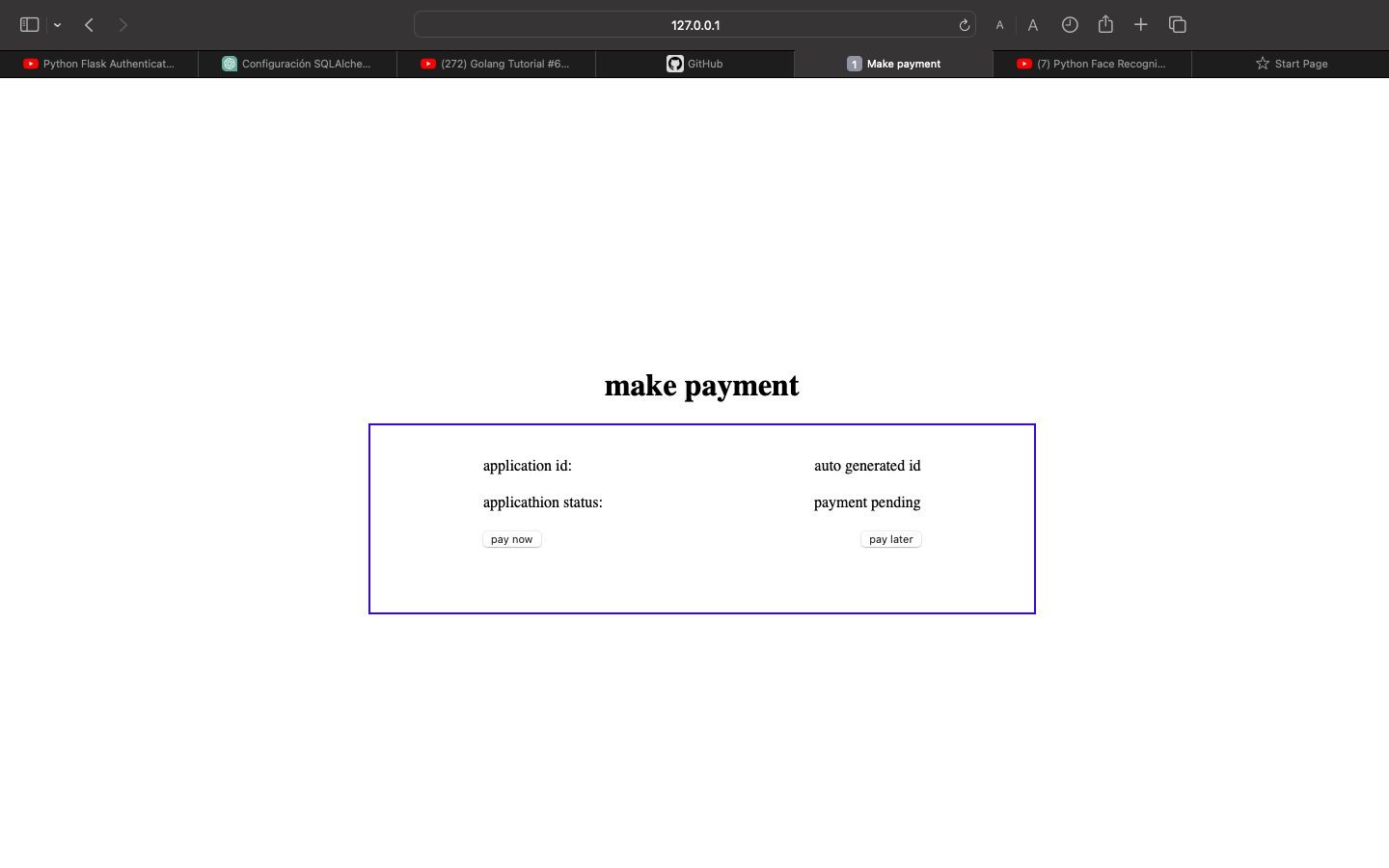
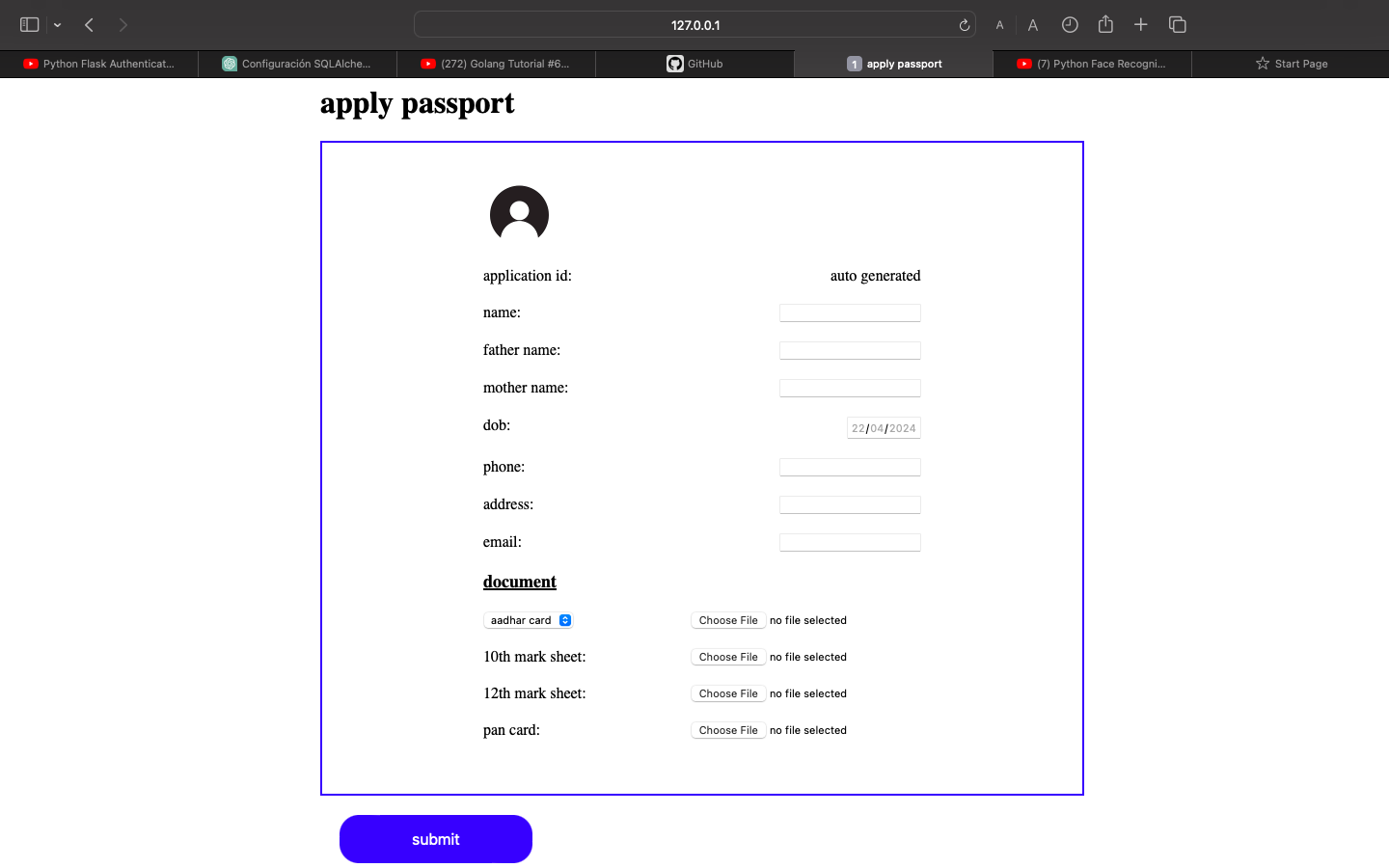
**OUTPUT**

****

****

****

****

****

Definitions:

Administrator - Refers to the super user who is the Central Authority who has been

vested with the privilege to manage the entire system. It can be any higher official in

the Regional Passport Office of Ministry of External Affairs.

• Applicant - One who wishes to obtain the Passport.

• PAS - Refers to this Passport Automation System.

• HTML - Markup Language used for creating web pages.

• HTTP - Hyper Text Transfer Protocol.

• TCP/IP – Transmission Control Protocol/Internet Protocol is the communication

Definitions:

Administrator - Refers to the super user who is the Central Authority who has been

vested with the privilege to manage the entire system. It can be any higher official in

the Regional Passport Office of Ministry of External Affairs.

• Applicant - One who wishes to obtain the Passport.

• PAS - Refers to this Passport Automation System.

• HTML - Markup Language used for creating web pages.

• HTTP - Hyper Text Transfer Protocol.

• TCP/IP – Transmission Control Protocol/Internet Protocol is the communication