TITLE:FISCALGUIDE

PROBLEM STATEMENT:INTEGRATED COMMON SERVICES TO COMMON PEOPLE

PARTICIPANT NAME:PADMAVATHI B
COLLEGE NAME:SRI SAI RAM INSTITUTE OF TECHNOLOGY
INTERNAL GUIDE:DR.SU.SUGANTHI

TOOLS USED:

L The fiscalguide project utilizes several tools and libraries to achieve its functionalities. Here's a breakdown of the key tools and libraries used:

1. Python Standard Libraries
   - Tkinter: Used for creating the graphical user interface (GUI).
     - Widgets such as Frame, Label, Entry, Button, Treeview are used for different components of the UI.
   - os: Used for file path manipulations.
   - datetime: Used for handling date and time, especially for tracking payment.

2)External Libraries
   - qrcode: Used for generating QR codes.
     - Generates QR codes for payments which users can scan for making payments.
   - Pillow (PIL): Used for handling image files.
     - Used in conjunction with the qrcode library to save and display QR code images.
   - Twilio: Used for sending SMS notifications.
     - Provides functionality to send SMS messages to users upon successful payments.
   - tkinter.ttk: A submodule of tkinter for advanced widgets.
     - Treeview widget is used for displaying tabular data such as bank interest rates and goal tracking.

3. APIs and Services
     - Twilio API:
     - Utilized for sending SMS notifications to users regarding payment confirmations or other alerts.

4. Integrated Development Environment (IDE)

- PyCharm, VS Code, or any other Python IDE:
  - Used for writing, testing, and debugging the Python code.

5. Version Control

- Git:
  - Used for version control and collaboration (not explicitly mentioned in the code but typically used in projects).

Using Tkinter for GUI:

```
import tkinter as tk
from tkinter import ttk
class FinancialNavigatorApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Financial Navigator")
        self.create_login_frame()

    def create_login_frame(self):
        self.login_frame = tk.Frame(self.root)
        self.login_frame.pack()

        self.username_label = tk.Label(self.login_frame, text="Username")
        self.username_label.grid(row=0, column=0)
        self.username_entry = tk.Entry(self.login_frame)
        self.username_entry.grid(row=0, column=1)
```

```
self.password_label = tk.Label(self.login_frame, text="Password")
self.password_label.grid(row=1, column=0)
self.password_entry = tk.Entry(self.login_frame, show="*")
self.password_entry.grid(row=1, column=1)

self.login_button = tk.Button(self.login_frame, text="Login", command=self.login)
self.login_button.grid(row=2, column=0, columnspan=2)

def login(self):
username = self.username_entry.get()
password = self.password_entry.get()
# Perform login validation
# On success, move to next frame
```

Generating QR Code:

```
import qrcode
from PIL import Image
def generate_qr_code(data, filename):
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )qr.add_data(data)
    qr.make(fit=True)mg = qr.make_image(fill='black', back_color='white') img.save(filename)
```

TECECHNOLOGIES USED

The financial navigator app leverages various technologies to deliver a comprehensive and efficient user experience. Here's a breakdown of the key technologies used:

1. Programming Language
   - Python: The primary programming language used for developing the app. Python's simplicity and extensive libraries make it suitable for both backend processing and frontend GUI development.
2. User Interface Development
   - Tkinter: A standard Python library for creating graphical user interfaces (GUI). Tkinter is used to design and implement the application's frontend, including login forms, data entry forms, and display tables.
3. Data Handling and Processing
   - Python Standard Libraries:
   - os: For file and directory operations.
   - datetime: For handling date and time operations, crucial for tracking payment dates and goal deadlines.
   - json: For reading and writing JSON data, useful for configuration and data persistence.
4. QR Code Generation
   - qrcode: A Python library used for generating QR codes. This is essential for the app's functionality to create QR codes for various tax and insurance payments.
   - Pillow (PIL): A Python Imaging Library (PIL) used alongside qrcode to handle image files and save QR code images.
5. Communication
   - Twilio: A cloud communications platform used for sending SMS notifications to users. This ensures users receive timely updates about their financial activities.

6)Data Storage and Management

- SQLite: A lightweight, disk-based database system integrated within Python for storing user data, payment records, and goal tracking information.

7)APIs and Web Services

- Twilio API: Used to send SMS notifications through the Twilio platform. This involves using REST API calls to send messages programmatically.

8)Integrated Development Environment (IDE)

- PyCharm, Visual Studio Code (VS Code), or any other Python IDE: These IDEs are used for writing, testing, and debugging the Python code. They provide features like code completion, syntax highlighting, and debugging tools.

9)Version Control

- Git: A version control system used for managing code changes, collaborating with team members, and maintaining version history.

10)Security

- Hashlib: A Python library for hashing passwords and ensuring secure login credentials.

To set up this project on your local machine, follow these steps:

Prerequisites

1. Python: Ensure you have Python 3 installed. You can download it from the official Python website.
2. pip: Ensure you have pip installed. It usually comes with Python, but you can install it separately if needed.

STEP 1:Create a Project Directory: Open your terminal or command prompt and create a directory for your project.

```
mkdir TaxPaymentApp
cd TaxPaymentApp
```

STEP 2:Create a Virtual Environment: It's a good practice to use a virtual environment for your project.

```
python -m venv venv
source venv/bin/activate  # On Windows use `venv\Scripts\activate`
```

STEP3:Install Required Packages: Install the necessary packages using pip.

```
pip install tkinter qrcode pillow twilio.
```

STEP 4:Download the Government Logo: Save a government logo image as government_logo.png in your project directory.

STEP 5:Create the Python Script: Create a Python file, e.g., app.py, and paste the code you provided into this file.

step 6:Twilio Configuration: Update the Twilio credentials and phone number in the TaxPaymentApp class with your actual Twilio account details. You need to sign up for a Twilio account and get an Account SID, Auth Token, and a Twilio phone number.

STEP 7:Run the Application: Run the Python script to start the application.
python app.py

Sample Directory Structure:
TaxPaymentApp/
|
├── venv/
├── government_logo.png
└── app.py

- Additional Notes:Ensure that government_logo.png is in the same directory as your Python script.
- If you encounter any issues with tkinter on some systems (especially Linux), you might need to install additional packages. For instance, on Ubuntu, you can use:
- sudo apt-get install python3-tk
- Running the Application:python app.py

ALGORITHMS USED:

1)User Authentication Algorithm:

- Login: This involves validating the username and password. The current implementation uses dummy authentication for simplicity.
- Check if username and password match predefined values.
- Display a message if authentication fails.
- Registration: Collect and store new user credentials. The current implementation uses dummy registration logic.
- Validate input fields (username and password).
- Store credentials (in practice, this should be securely handled).
- Password Reset: Allows users to reset their password. The current implementation uses dummy reset logic.
- Validate the username.
- Provide a method to reset the password (in practice, this should involve secure steps).

2)QR Code Generation Algorithm:

- For generating QR codes for payments.
- Collect payment details (tax/insurance type and amount).
- Use the qrcode library to generate a QR code.
- Save and display the QR code image.
- Libraries Used: qrcode and PIL (Pillow).

3)SMS Notification Algorithm:

- Send SMS notifications upon successful payment using Twilio.
- Initialize Twilio client with credentials.
- Use the Twilio client to send an SMS message with payment details.
- Library Used: twilio.

4) **UI Navigation Algorithm:**

- **Handles the transition between different sections of the app.**
- **Back and Next buttons to navigate through the different sections.**
- **Show and hide relevant frames based on the current section.**

5) Interest Rate Display Algorithm:

- **Display interest rates for different banks.**
- **Use Treeview to list interest rates.**
- **Currently, uses dummy data for interest rates.**

6) Goal Tracking Algorithm:

- **Manage and track financial goals.**
- **Add goals with specific details (goal name, amount, and progress).**
- **Display goals in a Treeview widget.**
- **Update goal progress and reflect changes in the display.**

THANK YOU