Reminder Application Documentation

# Project Overview

The Reminder Application is a web-based tool built using Flask that allows users to set reminders and receive email notifications. Users can input their reminder details through a form and generate a QR code for easy access to the form.

# Key Features

* **User Input Form**: Users can enter their username, email, reminder title, description, date, and time.
* **QR Code Generation**: Generates a QR code that links directly to the form, facilitating easy access.
* **Email Notifications**: Sends customized email notifications containing reminder details.

# Skills and Technologies Used

1. **Flask**

* The application is built on Flask, which manages routing and request handling.
* Example:

@app.route('/')

def qr\_code\_page():

return render\_template('qrcode.html')

1. **HTML/CSS**

* The user interface is designed with HTML and styled with CSS.
* Example (HTML form):

<form action="{{ url\_for('create\_reminder') }}" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<!-- Other fields... -->

</form>

1. **Python**

* The backend logic is implemented in Python.
* Example:

from datetime import datetime

reminder\_datetime = datetime.strptime(f"{reminder\_date} {reminder\_time}", '%Y-%m-%d %H:%M')

1. **Flask-Mail**

* Used for sending email notifications.
* Example: app.config['MAIL\_USERNAME'] = 'your\_email@gmail.com'

1. **QR Code Generation (qrcode library)**

* QR codes are generated to link to the reminder form.
* Example:

qr = qrcode.QRCode(version=1, box\_size=10, border=5)

qr.add\_data(url)

1. **Threading**

* Utilized to send emails without blocking the main application.
* Example:

threading.Thread(target=send\_email, args=(title, description, email, reminder\_datetime)).start()

1. **Logging**

* Implemented to capture application events and errors (to be tested).
* Example:

import logging

logging.basicConfig(filename='app.log', level=logging.INFO)

1. **User Feedback**

* Flash messages provide feedback on reminder submissions.
* Example:

flash('Reminder set successfully!', 'success')

**Use Cases Implemented So Far**

1. Users can set a reminder by filling out the form.
2. Users receive email notifications at the specified time.
3. Users can access the reminder form via a QR code.

**Additional Considerations**

* Explore deployment options to make the application accessible online.
* Enhance error handling and user experience based on feedback.

**Next Steps**

1. Test the logging functionality to ensure it works correctly.
2. Deploy the application on a hosting platform (e.g., Heroku).
3. Gather user feedback and iterate on the design and functionality.

**Nice to Add Features or Design**

1. **User Authentication**: Allow users to create accounts and manage their reminders securely.
2. **Database Integration**: Store reminders in a database for easy access and management.
3. **Responsive Design**: Improve the UI to be more mobile-friendly.
4. **Notification Options**: Add SMS or push notifications in addition to email reminders.
5. **Customization**: Allow users to customize reminder settings, such as snoozing.