Email Verification using Flask-Mail

1. Flask: This is the core framework used to build the web application.
2. Flask-Mail: This library is used to send emails from the Flask application.
3. itsdangerous: This library provides various functions to work with JSON Web Signatures (JWS) and TimeStamped Signatures (TSS), which are used to create and validate tokens in the code. Used to create a token that includes the user's email address and a timestamp. Then serialize the data into a token, and then load it back to deserialize the token back into the original data. We could set an expiration time of the signature eg: an hour.

Steps :

1. A user enters their email address into a form on the website.
2. The application generates a secure token that represents the user's email address using itsdangerous.
3. The token is embedded in a link, which is included in an email that is sent to the user.
4. The user clicks the link in their email, which directs them to a route in the application that includes the token as a parameter.
5. The application extracts the token from the URL, deserializes it back into the user's email address using itsdangerous, and checks that the token hasn't expired.
6. If the token is valid, the application marks the user's email address as confirmed in the database.

Password reset using Flask-Mail

1. Accept an email to verify the user.
2. If the email address is valid, we generate a token.
3. Now we send a mail to the user with an url containing our secret token.
4. After validating the token, we redirect the user to a page where he could overwrite his password and this password is updated in the database.