Deploying MYSQL with AWS:

A screenshot of a computer

Description automatically generated

Connecting to AWS MYSQL RDS Instance:

A screenshot of a computer

Description automatically generated

Creating a new user:

A screenshot of a computer

Description automatically generated with medium confidence

Users\_db on AWS reflecting the changes:

A screenshot of a computer

Description automatically generated

Two Factor Authentication Library. Secret Key will be updated before deployment. Using the current time but rounded down to the nearest 3 minutes. 2FA Codes will be unique every 3 minutes

A picture containing text, screenshot

Description automatically generated

Method that generates a unique six digit code that is valid for 3 minutes based on the username, the rounded time and a secret key:

A picture containing text, screenshot, font

Description automatically generated

Method that generates a hash from the username and the tfa code for comparison, if no tfa code is provided the current tfa code will be used:

A picture containing text, screenshot, font, software

Description automatically generated

Method that validates a login for a user based on their username and tfa\_code

A screen shot of a computer code

Description automatically generated with low confidence

Creating a basic login form for the API, testing the API by simply sending back the login info:

A screenshot of a login screen

Description automatically generated with medium confidence

Form is correctly hitting the API endpoint:

Login Process:

After many failed attempts first successful login:

A screenshot of a computer

Description automatically generated

User\_logins table functioning:

A screenshot of a computer

Description automatically generated with medium confidence

Login attempt with incorrect password:

A screenshot of a computer

Description automatically generated

Failed attempt saved in db:

A screenshot of a computer

Description automatically generated with medium confidence

Lockout Feature explanation and screenshots. When a user fails a login the failed\_login\_count column in the database increments (starts at 0). If the user fails 3 logins the following happens:

1. locked\_out column is set to 1 (default is 0)
2. locked\_out\_end is set to 5 minutes from now
3. failed\_login\_count resets to 0
4. All logins from the user will be blocked until the locked\_out\_end with a 403 code until time has passed

A screenshot of a computer

Description automatically generated

Database with an active lockout:

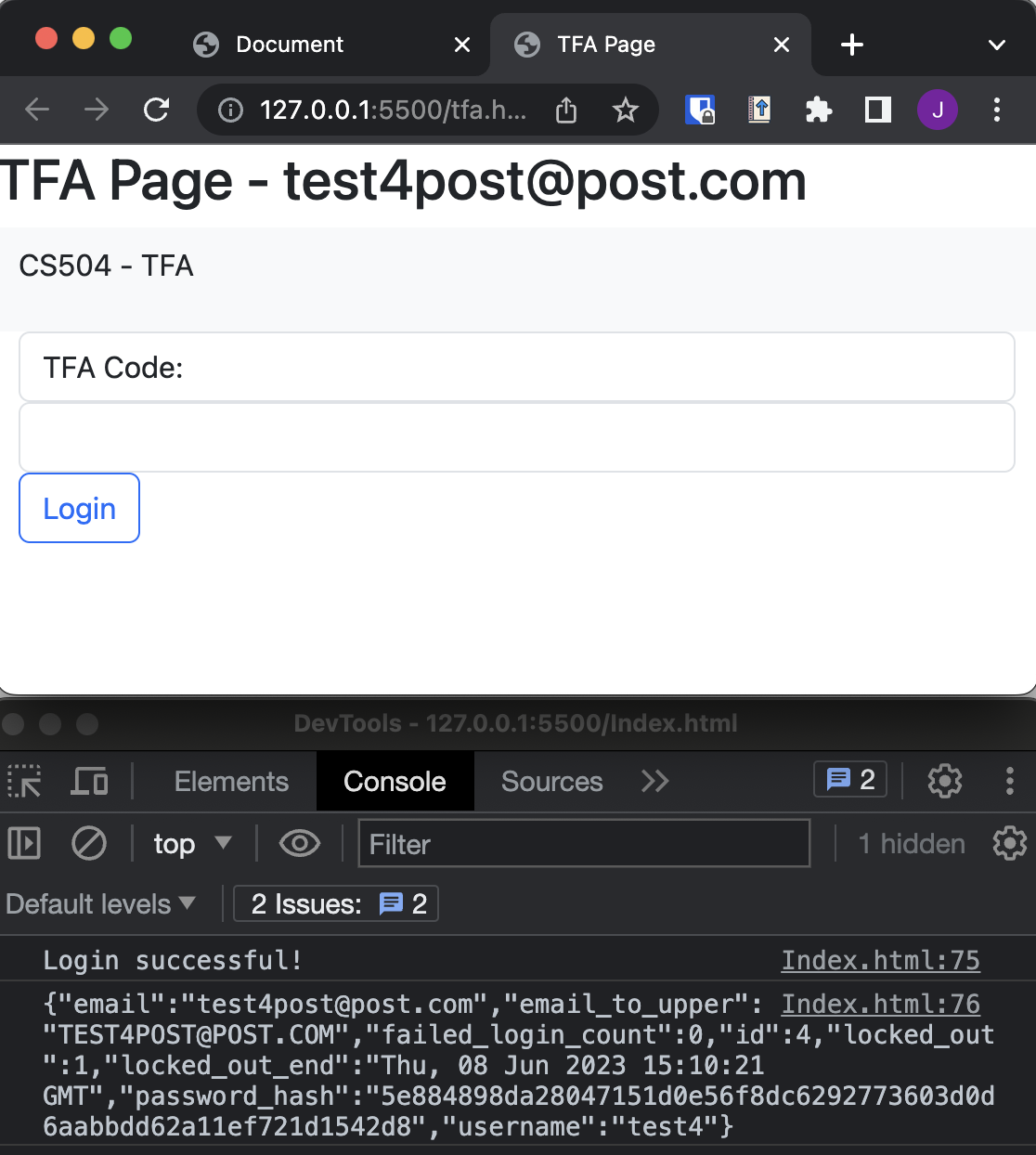
A screenshot of a computer

Description automatically generated

Successful login with redirect to 2FA page:

A screenshot of a computer

Description automatically generated with medium confidence



TFA\_Code generation with postman:

A screenshot of a computer

Description automatically generated

Code updates automatically with the time:

A screenshot of a computer

Description automatically generated

Testing for Validation:

A screenshot of a computer

Description automatically generated with medium confidence

Code has expired:

A screenshot of a computer

Description automatically generated with medium confidence

It finally works!

A screenshot of a computer

Description automatically generated

Adding SMS Service:

Unfortunately SMS is not working due to restrictions on the number that we are sending from:

For example this works and triggers on login:

A screen shot of a computer code

Description automatically generated with low confidence



However as soon as we try to concatenate the 2FA code it fails.

A screen shot of a computer code

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

The process to register our phone number would take 2-3 weeks. As of right now we do not have a way to send the user the 2FA Code.

For this reason I am including the 2FA code on another html page. We could easily implement SMS but we would need to go through this verification process first.

Here is the current complete login process using a test user on the live AWS database:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a message

Description automatically generated with low confidence