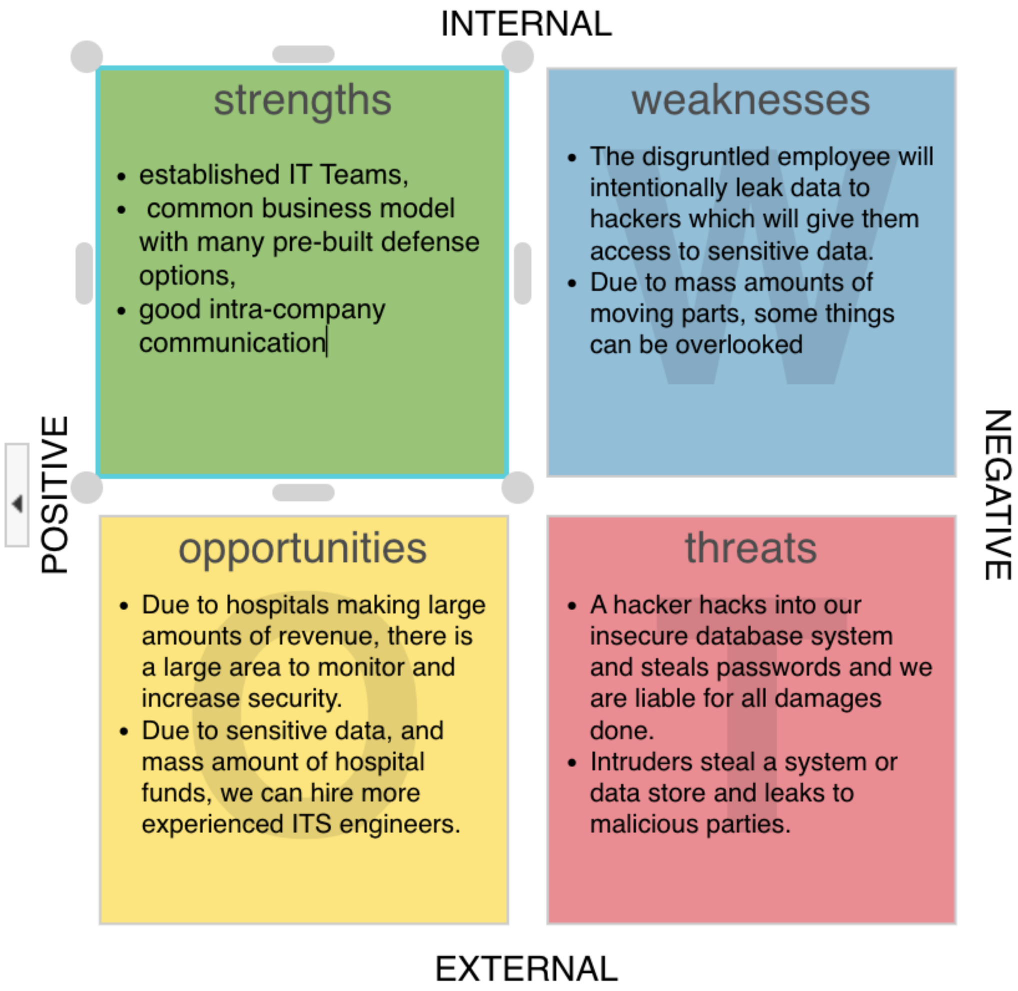
1. Summary
    - Within this project I determined major security risks presented, with these findings I have deemed a feasible fix for some and attempted to fix these risks.

1. Assessment Scope
    - I used tools such as Mac OS, visual studio code, github security features, and documentation tools such as google docs and excel. All of this was tested on a Safari browser.
    - Limitations faced with this assessment include lack of funding for proper testing tools, time, lack of a Windows machine were encountered.
2. Summary of Findings
    - In this assessment I found major issues including a lack of the AAA methodology implemented into the code itself such as logging of user login attempts. Another major findings included lack of any processes such as security and standard hospital documentation. Along with lack of documentation the web app does not include any ISO standard for handling user information. Finally, another major finding was that the repository where code is hosted (github) was not private and did not implement any of the github security features provided. Though these are the major findings there are still a multitude of security risks present.

Figure 1:



strengths

- established IT Teams,
- common business model with many pre-built defense options,
- good intra-company communication

weaknesses

- The disgruntled employee will intentionally leak data to hackers which will give them access to sensitive data.
- Due to mass amounts of moving parts, some things can be overlooked

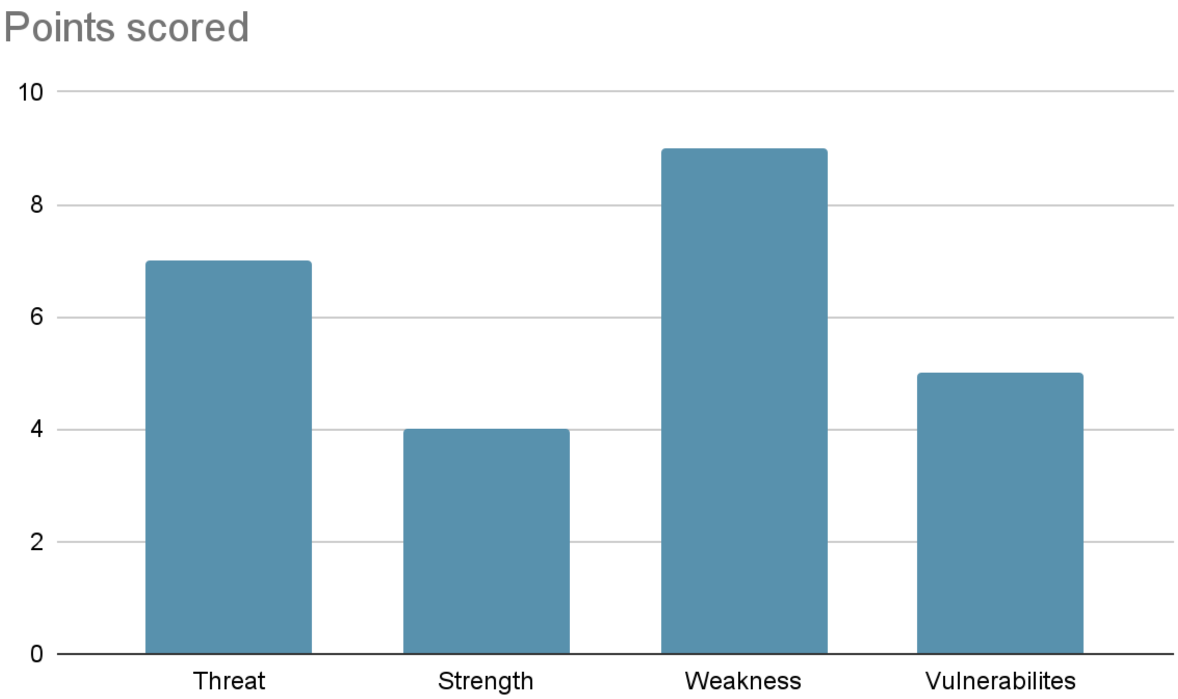
opportunities

- Due to hospitals making large amounts of revenue, there is a large area to monitor and increase security.
- Due to sensitive data, and mass amount of hospital funds, we can hire more experienced ITS engineers.

threats

- A hacker hacks into our insecure database system and steals passwords and we are liable for all damages done.
- Intruders steal a system or data store and leaks to malicious parties.

Figure 2: (How many updated types from assessment)



3. Summary of Recommendations
    - Changes made to the project were additional logging for user login attempts, security documentation added into the project, new procedures, and github security features enabled. The changes still needed are security to the hosted database, more in depth procedures that comply with ISO standards, and additional logging to report more unusual activity to a required security admin. (Information gathered through in class presentations)

2. Goals, Findings, and Recommendations

1. Assessment Goals
    - The purpose of this assessment was to determine any major risks found in the Hospital Management System in which an attacker could manipulate data.

2. Detailed Findings

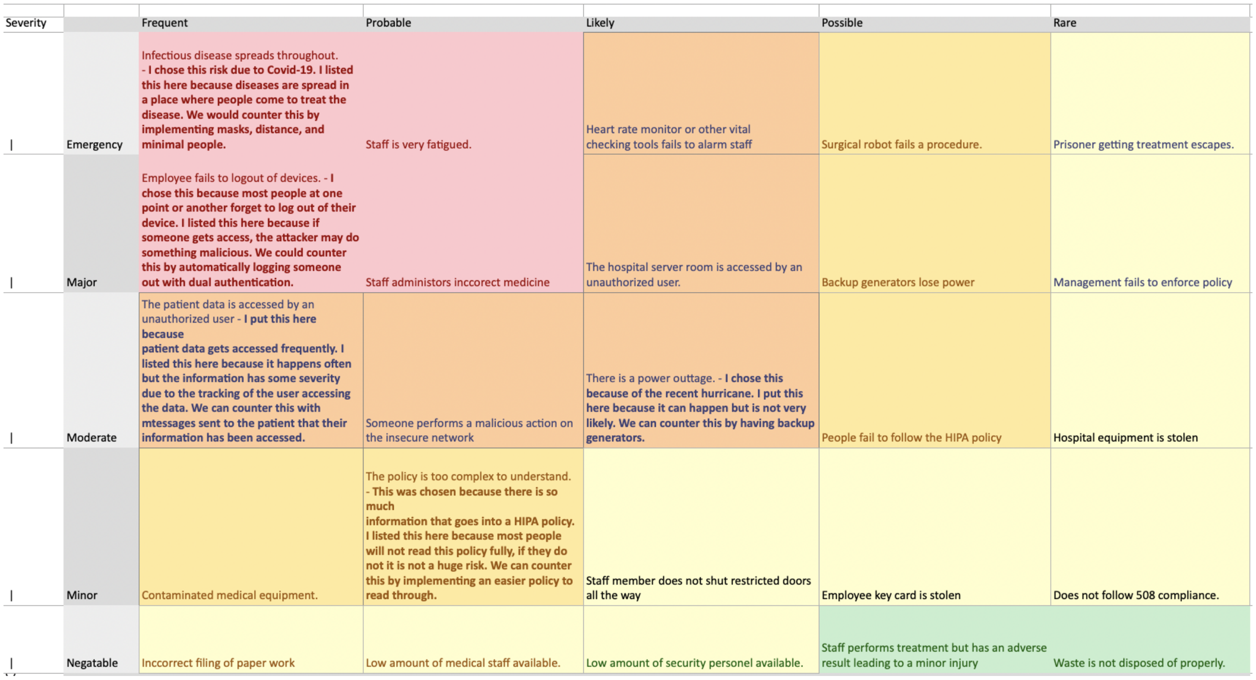
Type (Threat, Weakness, Vulnerability)	How (TYPE) Was Determined
Disgruntled Employee - Weakness (From Lecture)	On the occasion this occurs a disgruntled employee with unauthorized access is identified as an internal vulnerability.
A vital feature being overlooked - Weakness	Due to a mass web app such as this hospital management system a vital feature or feature can be overlooked.
A patient accesses a feature that was unintended for access - Weakness	The system should go through vigorous testing to ensure quality and prevention of unintentional unauthorized access.
A backdoor being left unsecure - threat (From lecture)	The system and all backdoors should be properly closed and secure to prevent an intruder of gaining access.
All repositories should be updated to latest update - Weakness	All dependencies should be updated daily if possible to ensure security features are always up to date.
A hacker gains access to the database - Threat (Try Hack Me)	If the connection string is available or the API connection is not properly handled, the intruder will gain access to confidential information.
An intruder gains access through the web app and manipulates data intentionally - Threat (Try Hack Me)	The system should ensure proper security through constant testing by security professionals.

3. Recommendations

Types From Figure 2.1	Recommendations
Disgruntled employee	Ensure the employee has been revoked access the moment they have been let go from the company. Also ensure logging is in place to determine suspicious activity.
Vital feature overlooked, backdoor	Set proper logging and standards for code as well as testing.
Unauthorized access through web app	Ensure logging is in place, proper testing, and code standards.
Repos up to date	Set up automatic updates for the repo and have it directly pushed through automatic CLI
Unauthorized DB access	Ensure endpoints are secure with logging and a security team to test for vulnerabilities.

3. Methodology for the Security Assessment

1. Risk Assessment Accuracy



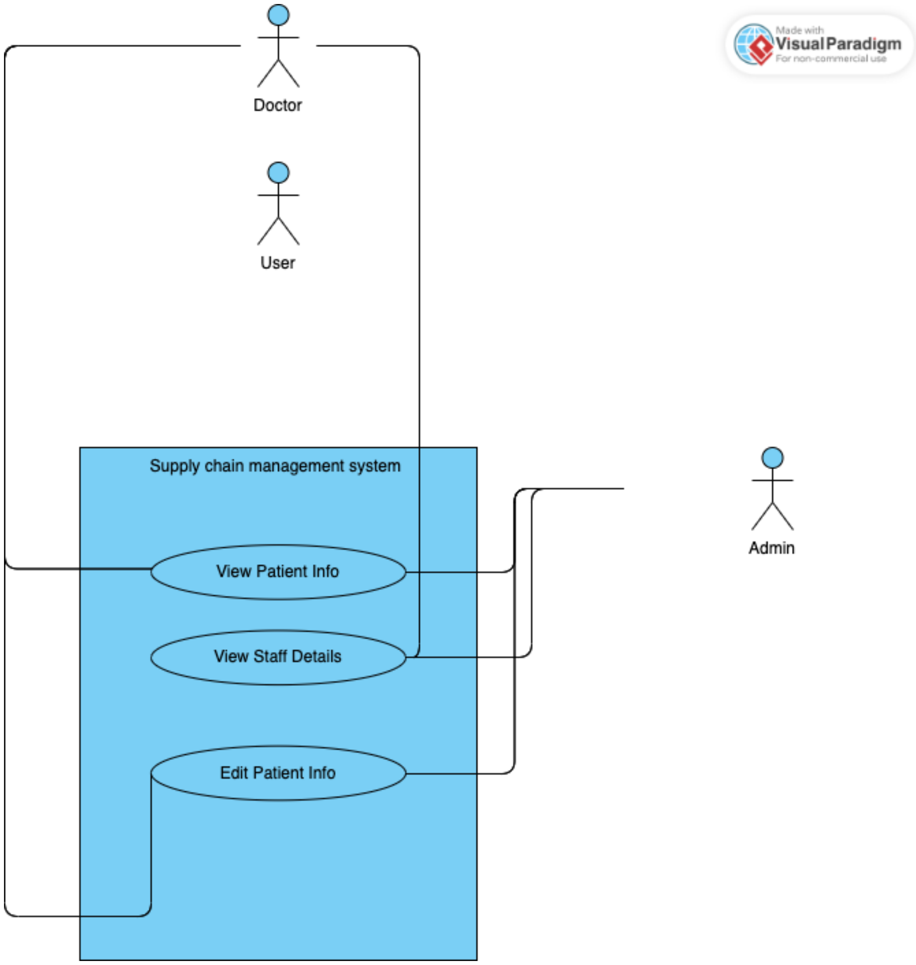
Severity		Frequent	Probable	Likely	Possible	Rare
|	Emergency	Infectious disease spreads throughout. **- I chose this risk due to Covid-19. I listed this here because diseases are spread in a place where people come to treat the disease. We would counter this by implementing masks, distance, and minimal people.**	Staff is very fatigued.	Heart rate monitor or other vital checking tools fails to alarm staff	Surgical robot fails a procedure.	Prisoner getting treatment escapes.
|	Major	Employee fails to logout of devices. - **I chose this because most people at one point or another forget to log out of their device. I listed this here because if someone gets access, the attacker may do something malicious. We could counter this by automatically logging someone out with dual authentication.**	Staff administors inccorect medicine	The hospital server room is accessed by an unauthorized user.	Backup generators lose power	Management fails to enforce policy
|	Moderate	The patient data is accessed by an unauthorized user - **I put this here because patient data gets accessed frequently. I listed this here because it happens often but the information has some severity due to the tracking of the user accessing the data. We can counter this with mtessages sent to the patient that their information has been accessed.**	Someone performs a malicious action on the insecure network	There is a power outtage. - **I chose this because of the recent hurricane. I put this here because it can happen but is not very likely. We can counter this by having backup generators.**	People fail to follow the HIPA policy	Hospital equipment is stolen
|	Minor	Contaminated medical equipment.	The policy is too complex to understand. **- This was chosen because there is so much information that goes into a HIPA policy. I listed this here because most people will not read this policy fully, if they do not it is not a huge risk. We can counter this by implementing an easier policy to read through.**	Staff member does not shut restricted doors all the way	Employee key card is stolen	Does not follow 508 compliance.
|	Negatable	Inccorrect filing of paper work	Low amount of medical staff available.	Low amount of security personel available.	Staff performs treatment but has an adverse result leading to a minor injury	Waste is not disposed of properly.

Of this assessment I found that most of my risk assessments were not accurate or did not apply to IT/Info Sec. Some of the ones that were accurate are the assessments dealing with policies and physical security of devices/technology.

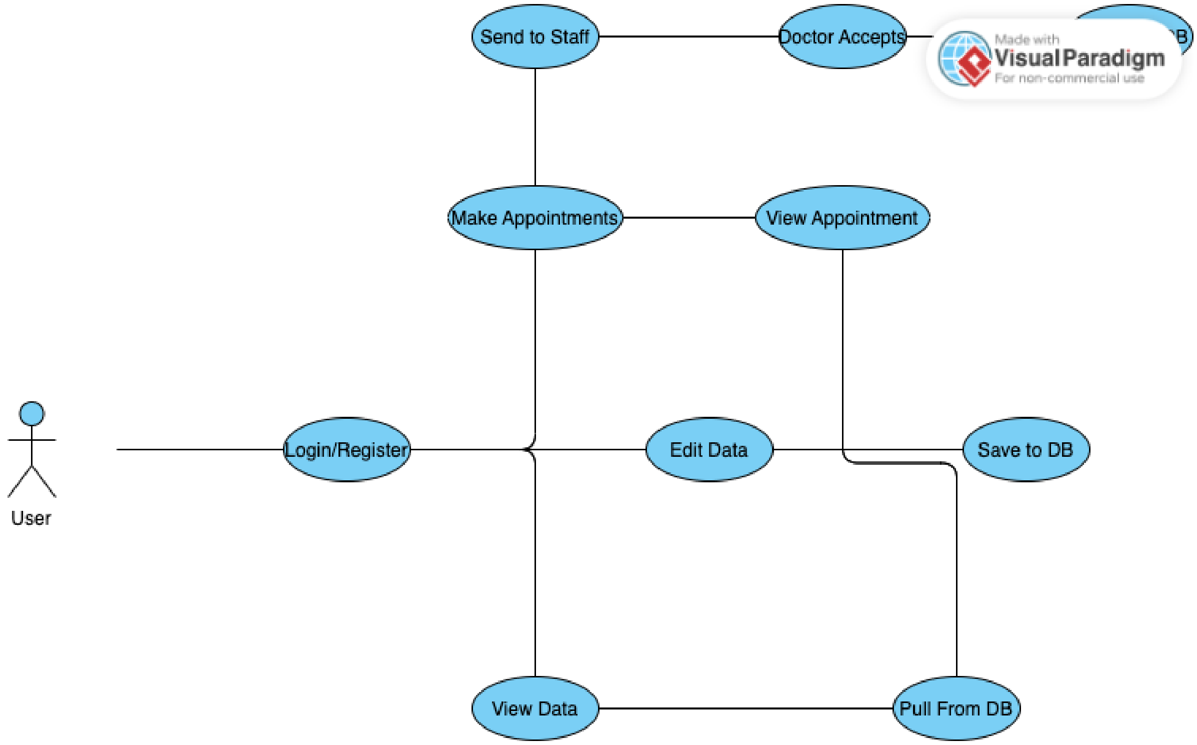
2. Risk Assessment Elements
    - Tools Used and Purpose: The tools used to determine these risks in the risk assessment are malware devices planted in the hospital to perform manipulation of data.
    - Pen Testing: I would perform pen testing on hospital management system to access unauthorized data
    - Analysis of Test Results: Outdated components (OWASP top 10)

4. Figures and Code

1. Figure 4.1



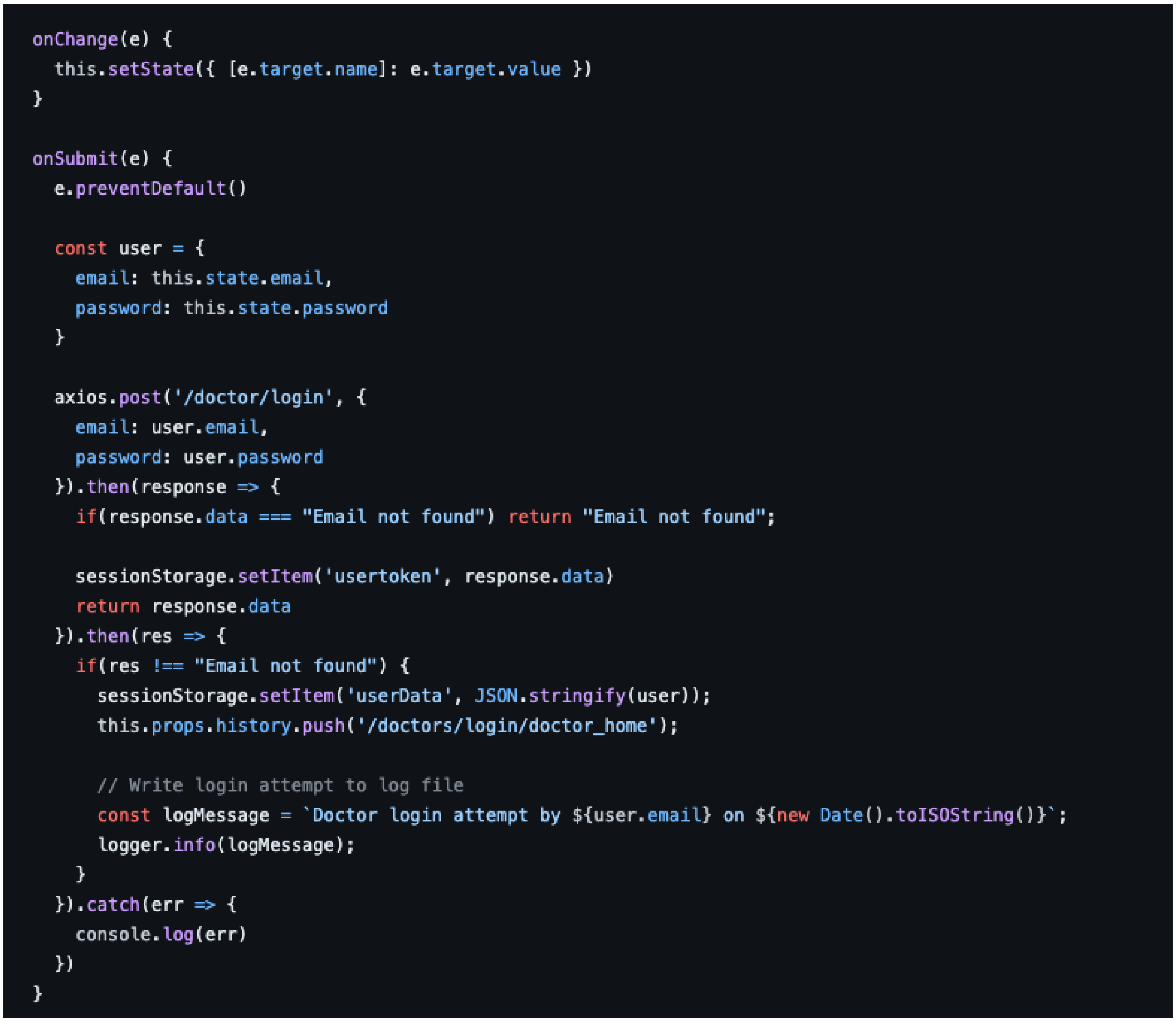
2. Figure 4.2



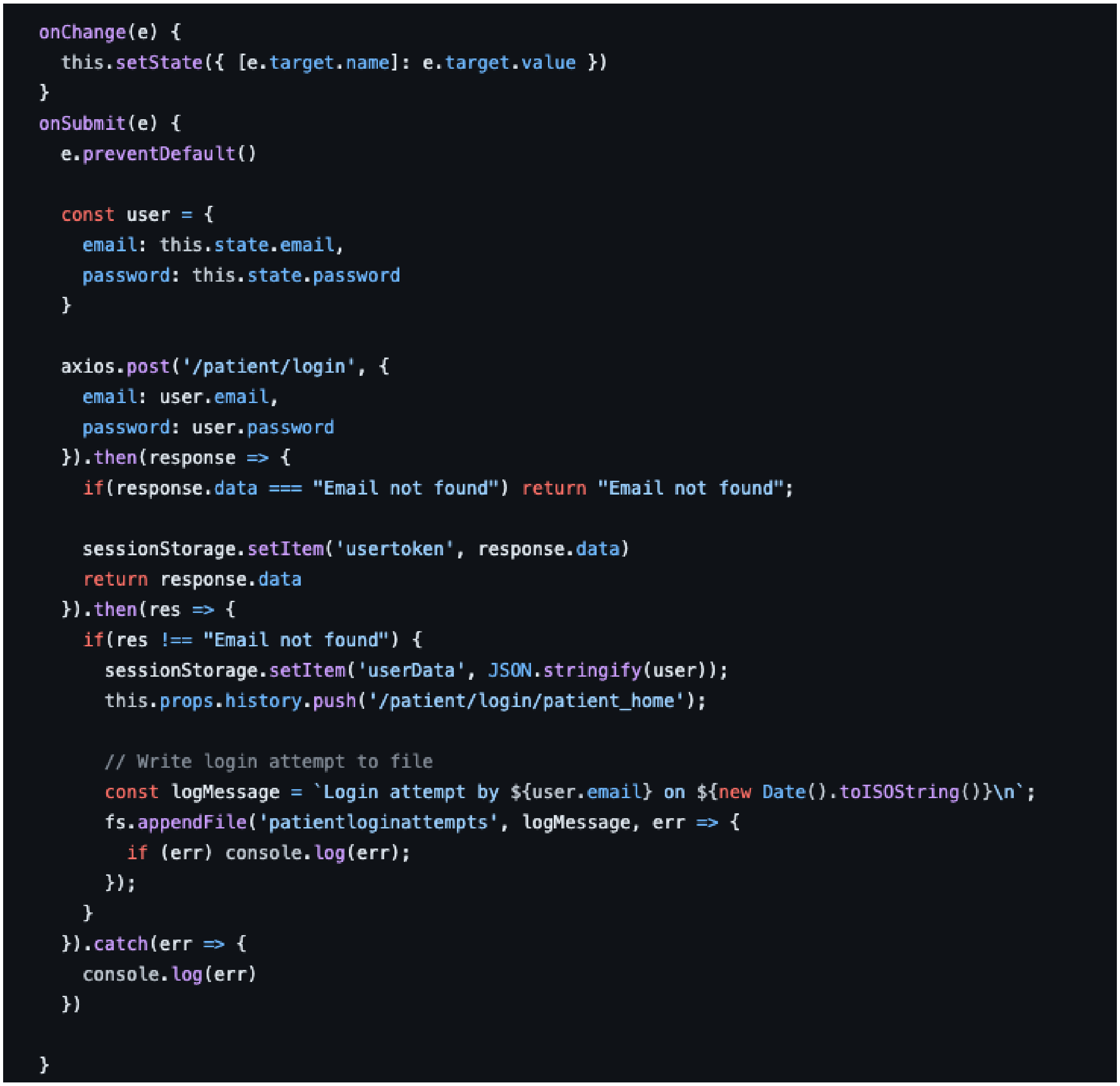
2. Code

Added updated to dependencies
Provided changes to github settings to increase security
Added documentation for processes

Github Code:



```
onChange(e) {
  this.setState({ [e.target.name]: e.target.value })
}

onSubmit(e) {
  e.preventDefault()

  const user = {
    email: this.state.email,
    password: this.state.password
  }

  axios.post('/doctor/login', {
    email: user.email,
    password: user.password
  }).then(response => {
    if(response.data === "Email not found") return "Email not found";

    sessionStorage.setItem('usertoken', response.data)
    return response.data
  }).then(res => {
    if(res !== "Email not found") {
      sessionStorage.setItem('userData', JSON.stringify(user));
      this.props.history.push('/doctors/login/doctor_home');

      // Write login attempt to log file
      const logMessage = `Doctor login attempt by ${user.email} on ${new Date().toISOString()}`;
      logger.info(logMessage);
    }
  }).catch(err => {
    console.log(err)
  })
}
```



```
onChange(e) {
  this.setState({ [e.target.name]: e.target.value })
}
onSubmit(e) {
  e.preventDefault()

  const user = {
    email: this.state.email,
    password: this.state.password
  }

  axios.post('/patient/login', {
    email: user.email,
    password: user.password
  }).then(response => {
    if(response.data === "Email not found") return "Email not found";

    sessionStorage.setItem('usertoken', response.data)
    return response.data
  }).then(res => {
    if(res !== "Email not found") {
      sessionStorage.setItem('userData', JSON.stringify(user));
      this.props.history.push('/patient/login/patient_home');

      // Write login attempt to file
      const logMessage = `Login attempt by ${user.email} on ${new Date().toISOString()}\n`;
      fs.appendFile('patientloginattempts', logMessage, err => {
        if (err) console.log(err);
      });
    }
  }).catch(err => {
    console.log(err)
  })

}
```

5. Works Cited

1. Lecture 1 First week
2. Lecturer 2 several weeks into the semester
3. TryHackMe - Pen Testing
4. TryHackMe - OWASP
5. https://owasp.org/Top10/
6. Course presentations