Individual Reflective

Secure Systems Architecture module covered a wide range of beneficial topics like:

- Introduction to Operating Systems and Secure Systems Architecture
- Modelling and Socket Programming
- Systems Engineering and Modelling
- The Scientific Method and its Role in Modelling Distributed Services and Systems
- Current and Future Challenges of Operating Systems and Distributed Systems
- Security Testing of Distributed Systems

Attached is the document "Evidence" which displays screenshots of my module contributions i.e., Initial post, PowerPoint presentation, project's python scripts, etc.

Additionally, I had the opportunity to collaborate on a group project with three of my peers. Our selected case study, (Kodali, et al., 2016), focused on a low-cost system serving as a smart home security and home automation system. Our task was to create a detailed proposal/design document (in unit 3), that would guide the development of the system.

To ensure project's success, we appointed a group leader and agreed on matters like: communication methods (Google Drive, Zoom, WhatsApp, emails, GitHub), work distribution, meeting schedules, etc.

The first part of the project, design document creation, I identified the case study risks, potential vulnerabilities, mitigations, and designed UML diagrams. Evidence doc: Figures 3-10, is a compilation of my findings from sources such as Abdullah et al.

(2019), Anand et al. (2020), etc. This increased my understanding of home security and automation systems, including IoT devices, and their potential risks, such as dependence on Wi-Fi, susceptibility to Denial-of-Sleep (DoSL) attacks, the effectiveness of bandwidth limitations, among others.

I faced a challenge running an AD tool with Java. During one of our discussions, I requested help from my team and one of them remotely guided me using TeamViewer.

The second part of the project, we created a python prototype system that demonstrated the functioning and interaction between a simulated device and a simulated controller, so as to investigate one of our hypothesis questions.

We faced a challenge due to our lack of coding experience, so we decided to first research on it and held regular meetings (every 1-3 days) to share our findings. Initially, we opted to use Django to design an application with 2-Factor Authentication login, user registration and authorisation. After my peer completed the Django app, I created a Docker image and container, and scanned it (Evidence doc: Figures 11-15).

However, after further research on IoT devices, since our app was not simulating any of the home automation controls, I discovered that paho-mqtt open-source client libraries for Python were more suitable for machine-to-machine communications (PyPi, N.D.). I proposed this to my group and after their own research, we agreed to create a light control prototype using paho-mqtt and then dockerized it later.

I designed a motion-simulated light control and performed manual and automatic tests using pytest (Evidence doc: Figure 16), while my peers worked on other simulations, security features, testing, dockerization, and documentation.

Although we did not investigate my hypothesis question "*does the motion sensor device and light control system function collaboratively to ensure secure communication and operation of the system?"*, Figures 17-18 in the Evidence document show a successful connection return code (rc=0) using the MQTT protocol and a broker to ensure reliable and secure message delivery. Additionally, the use of JSON encoding ensures easy readability and understanding of the messages between the devices (Steve, 2022).

However, we chose to investigate my peer's hypothesis on authentication because it provided in-depth analysis of one of the security features.

In terms of the development cycle, we followed an Agile development methodology and I believe that we were able to work on the project in an organized, flexible way and stayed on track and met our deadlines. At the same time, we could have used our time well by not focusing too much on one task. While in testing, we were able to perform both manual and automated tests on the connectivity, subscription, security, etc, - of the simulation so as to ensure that it was functioning as expected.

Some of the biggest challenges that we faced as a team were time constraints and the lack of enough learning materials on paho-mqtt on a Windows environment, therefore making it difficult to successfully implement some of the security features including the ones we had planned to implement in the project.

Despite of these difficulties, our collaborative and hardworking group was able to implement key features, resulting in a functional project submission.

Generally, the whole experience has been very knowledgeable and eye-opening. Moreover, I recognized the importance of good communication and collaborating with team members so as to ensure that we were all working towards the same goals.

Moving forward, I will use the knowledge and experience gained from this project to continue improving my technical and communication skills. I will also apply the lessons learned from this experience to future projects, ensuring that I can contribute effectively to development teams and deliver high-quality products that meet the set requirements.

References

Abdullah, T., Ali, W., Malebary, S. & Ahmed, A. A., 2019. A Review of Cyber Security Challenges, Attacks and Solutions for Internet of Things Based Smart Home. *International Journal of Computer Science and Network Security (IJCSNS),* 19(9), pp. 139-146.

Anand, P. et al., 2020. IoT Vulnerability Assessment for Sustainable Computing: Threats, Current Solutions, and Open Challenges. *IEEE Access,* Volume 8, pp. 168825-168853.

Borgini, J., 2021. *Tackle IoT application security threats and vulnerabilities.* [Online] Available at: https://www.techtarget.com/iotagenda/tip/Tackle-IoT-application-security-threats-and-vulnerabilities
[Accessed 2 February 2023].

Kodali, R. K., Jain, V., Bose, S. & Boppana, L., 2016. *IoT based smart security and home automation system.* Greater Nodia, IEEE.

PyPi, N.D.. *paho-mqtt 1.6.1.* [Online]
Available at: https://pypi.org/project/paho-mqtt/#:~:text=This%20code%20provides%20a%20client,.9%2B%20or%203.6%2B.
[Accessed 26 February 2023].

Steve, 2022. *Python MQTT Client Connections– Working with Connections.* [Online]
Available at: http://www.steves-internet-guide.com/client-connections-python-mqtt/
[Accessed 10 February 2023].

Screenshots of my Contributions in this Module

Figure 1: PowerPoint Presentation on Symbian Operating System

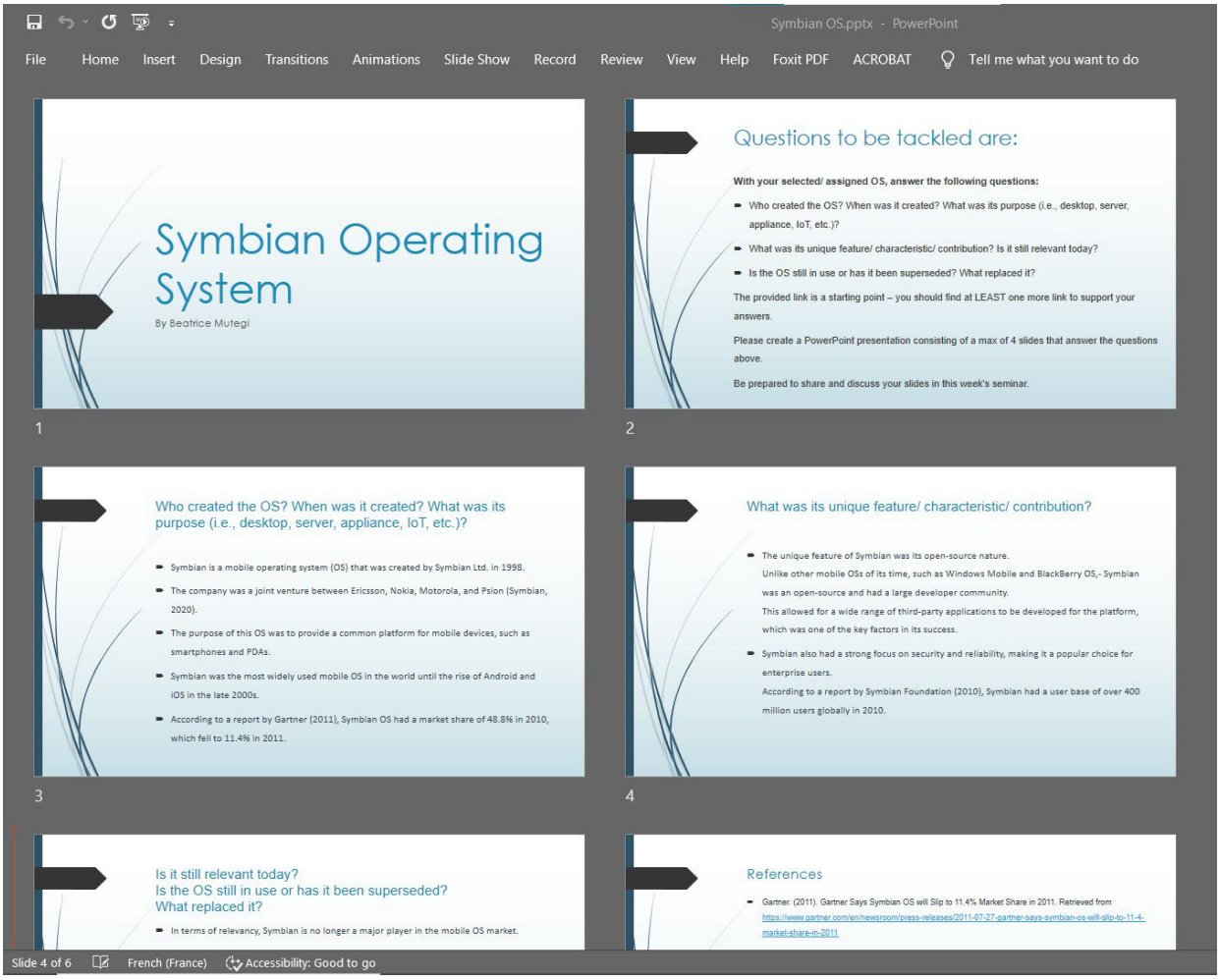


Figure 2: Initial Post

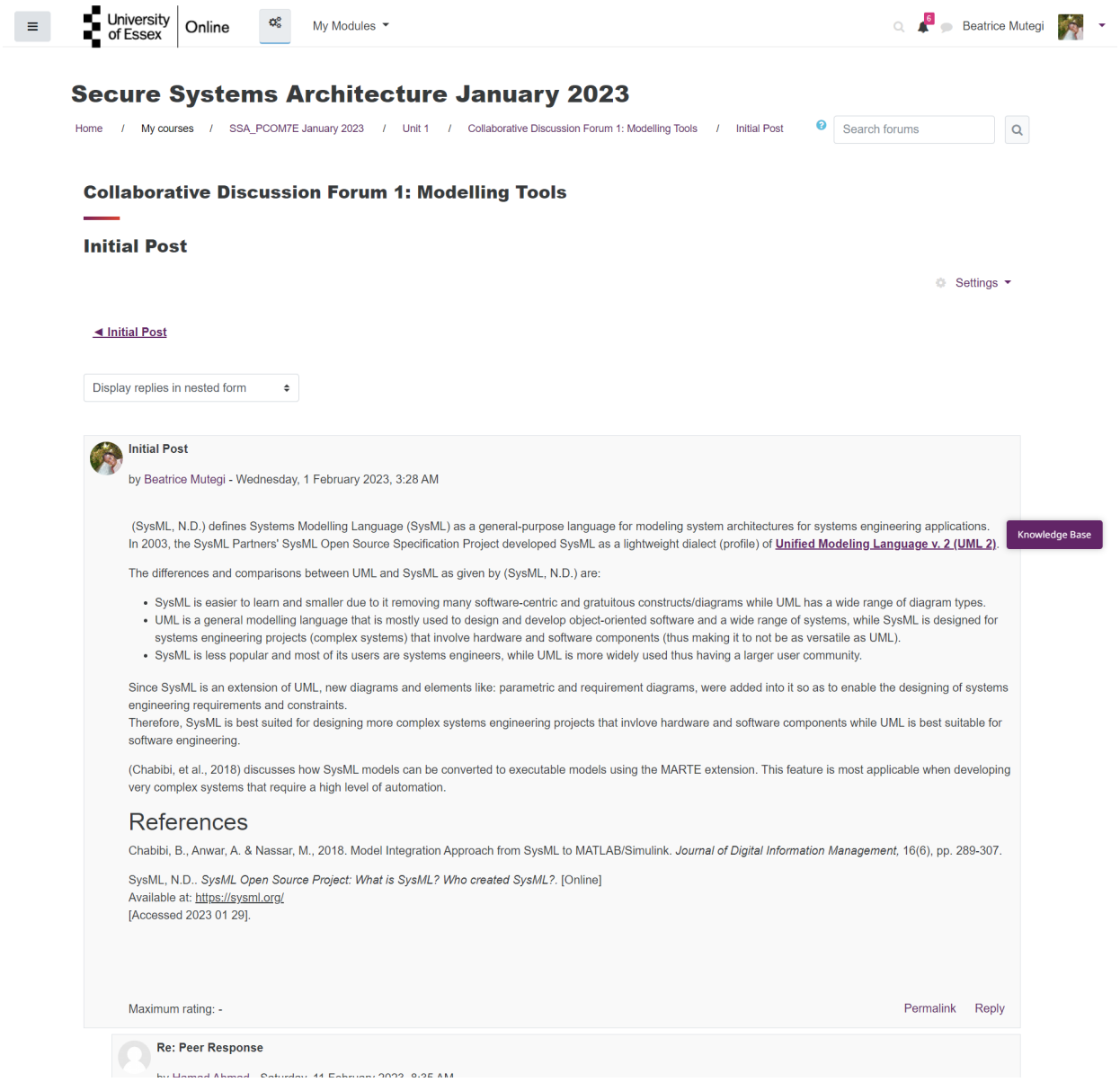
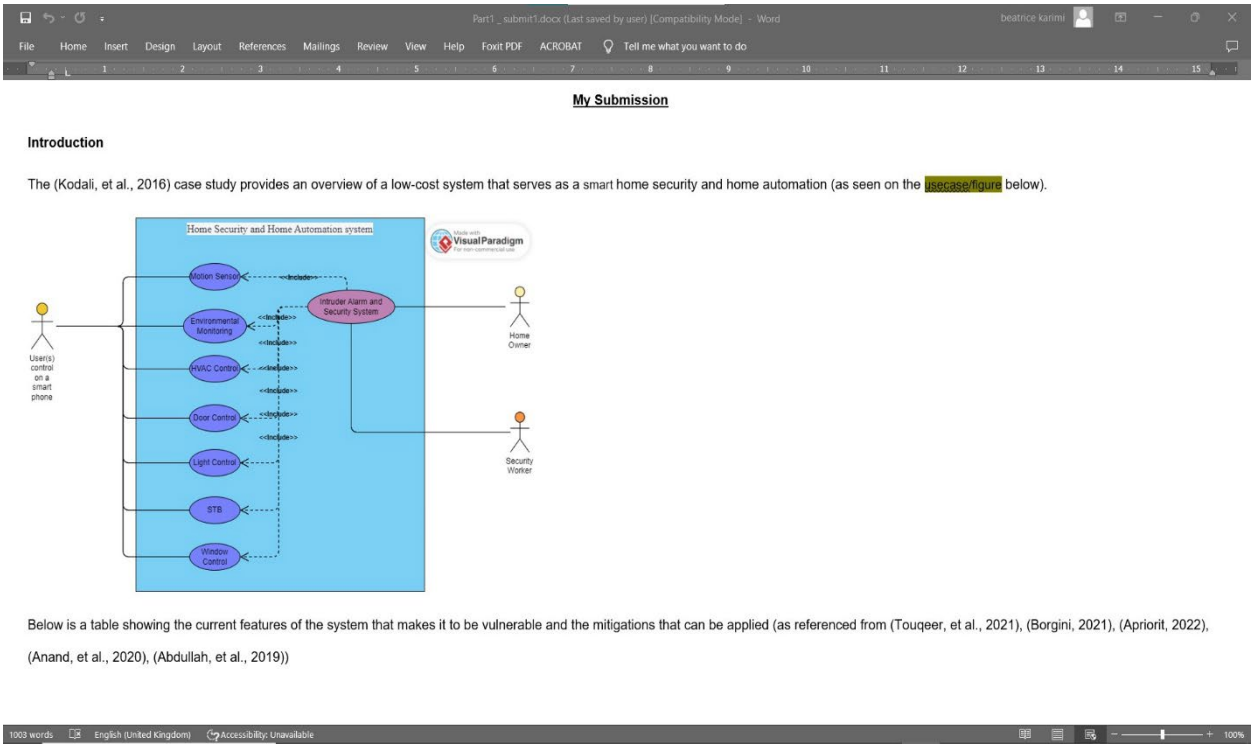


Figure 3: My Contribution to 1st Part of the Group Assessment (A)



Part1 _ submit1.docx (Last saved by user) [Compatibility Mode] - Word

beatrice karimi

File Home Insert Design Layout References Mailings Review View Help Foxit PDF ACROBAT Tell me what you want to do

My Submission

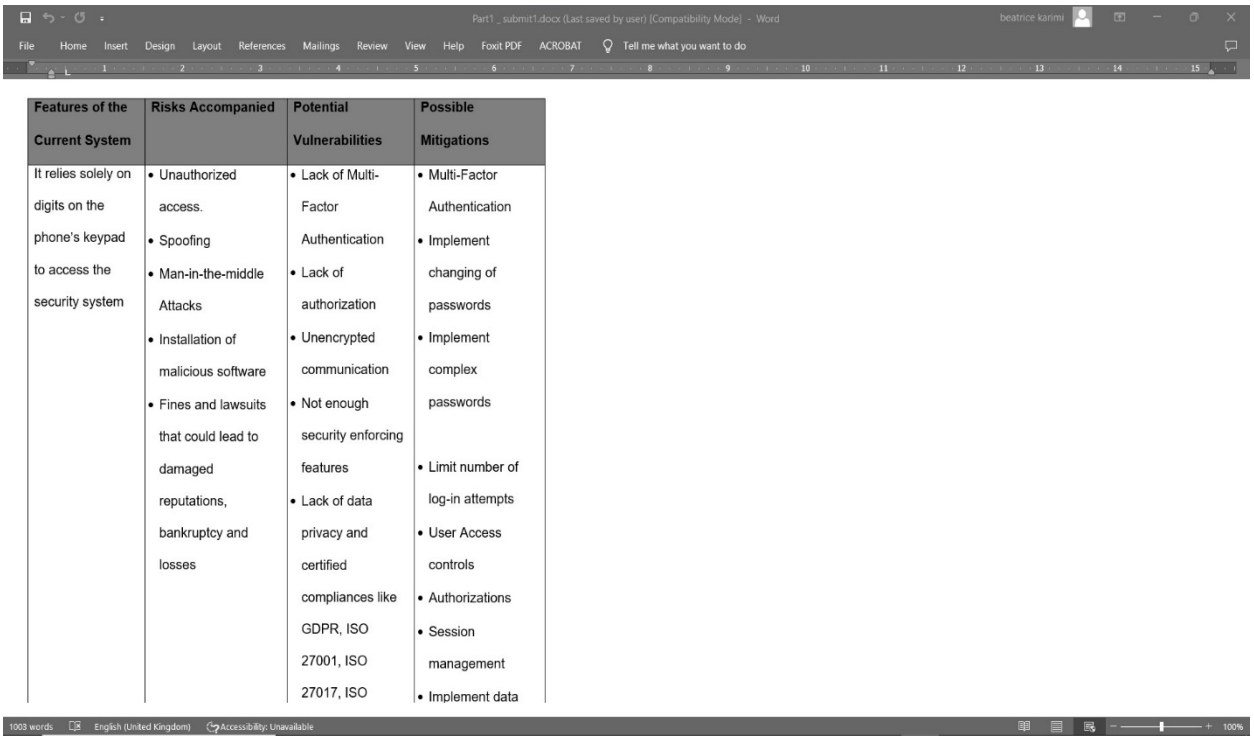
Introduction

The (Kodali, et al., 2016) case study provides an overview of a low-cost system that serves as a smart home security and home automation (as seen on the usecase/figure below).

Below is a table showing the current features of the system that makes it to be vulnerable and the mitigations that can be applied (as referenced from (Touqeer, et al., 2021), (Borgini, 2021), (Apriorit, 2022), (Anand, et al., 2020), (Abdullah, et al., 2019))

1003 words English (United Kingdom) Accessibility: Unavailable 100%

Figure 4: My Contribution to 1st Part of the Group Assessment (B)



Part1 _ submit1.docx (Last saved by user) [Compatibility Mode] - Word

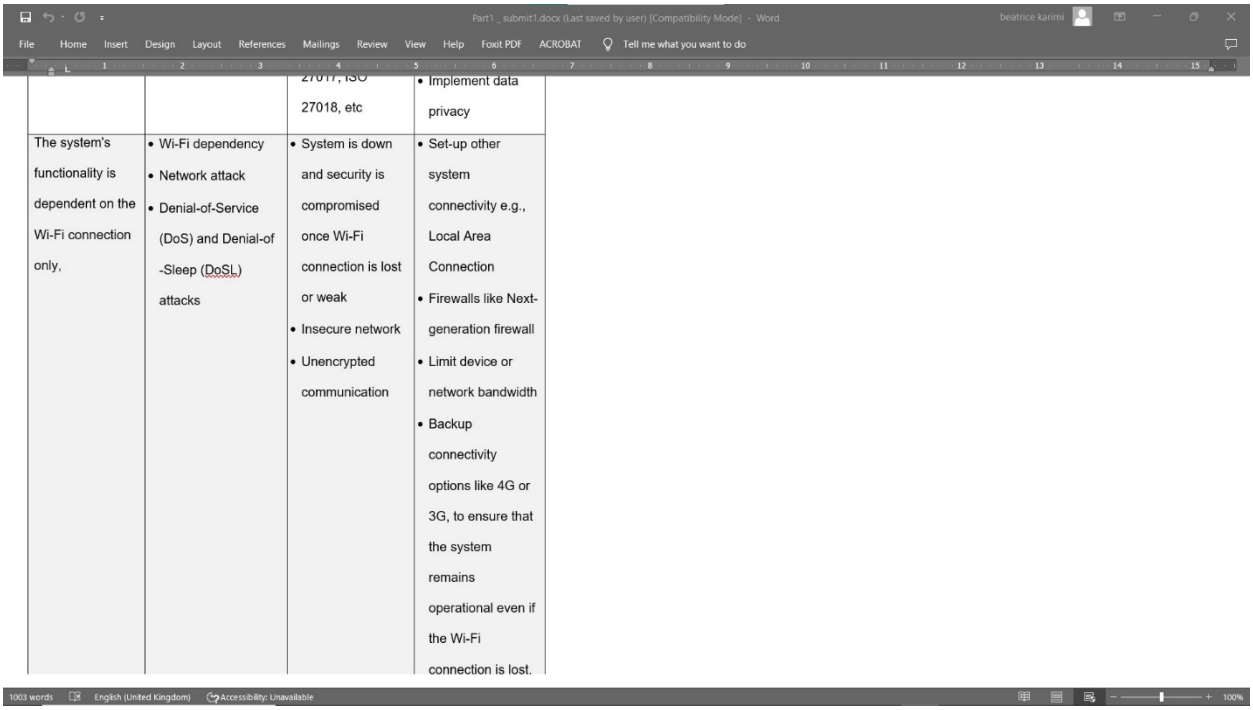
beatrice karimi

File Home Insert Design Layout References Mailings Review View Help Foxit PDF ACROBAT Tell me what you want to do

Features of the Current System	Risks Accompanied	Potential Vulnerabilities	Possible Mitigations
It relies solely on digits on the phone's keypad to access the security system	• Unauthorized access. • Spoofing • Man-in-the-middle Attacks • Installation of malicious software • Fines and lawsuits that could lead to damaged reputations, bankruptcy and losses	• Lack of Multi-Factor Authentication • Lack of authorization • Unencrypted communication • Not enough security enforcing features • Lack of data privacy and certified compliances like GDPR, ISO 27001, ISO 27017, ISO	• Multi-Factor Authentication • Implement changing of passwords • Implement complex passwords • Limit number of log-in attempts • User Access controls • Authorizations • Session management • Implement data

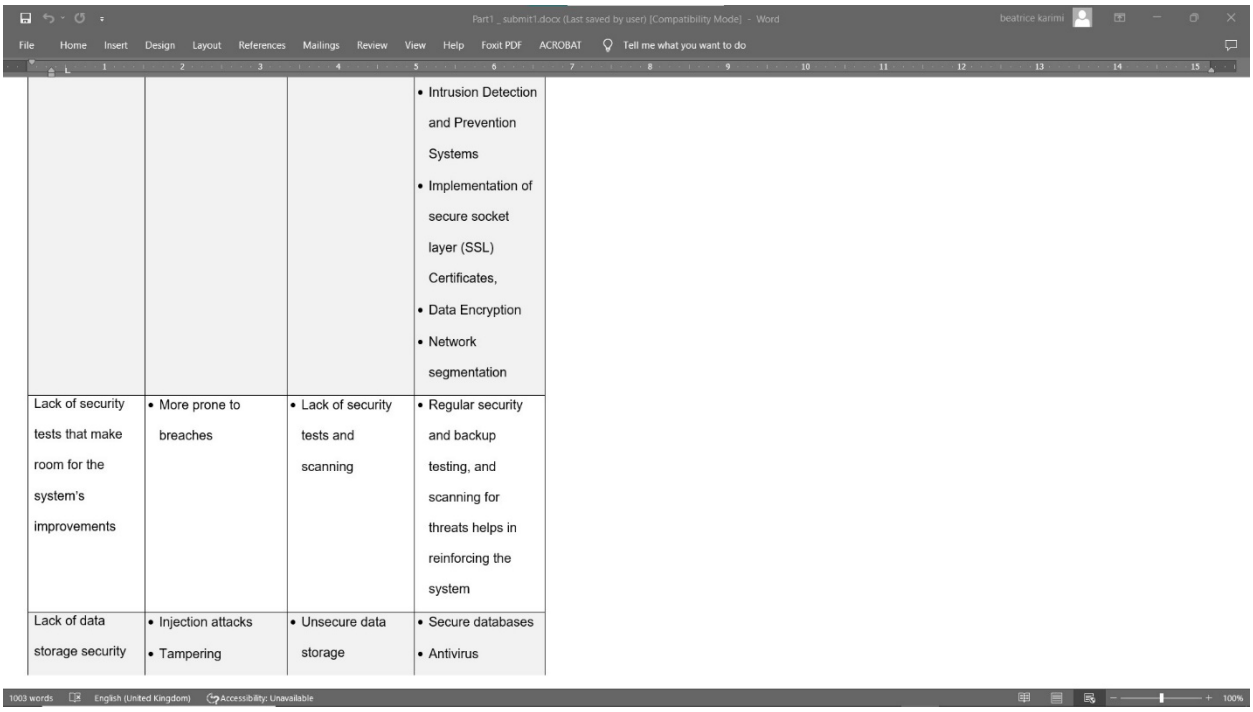
1003 words English (United Kingdom) Accessibility: Unavailable 100%

Figure 5: My Contribution to 1st Part of the Group Assessment (C)



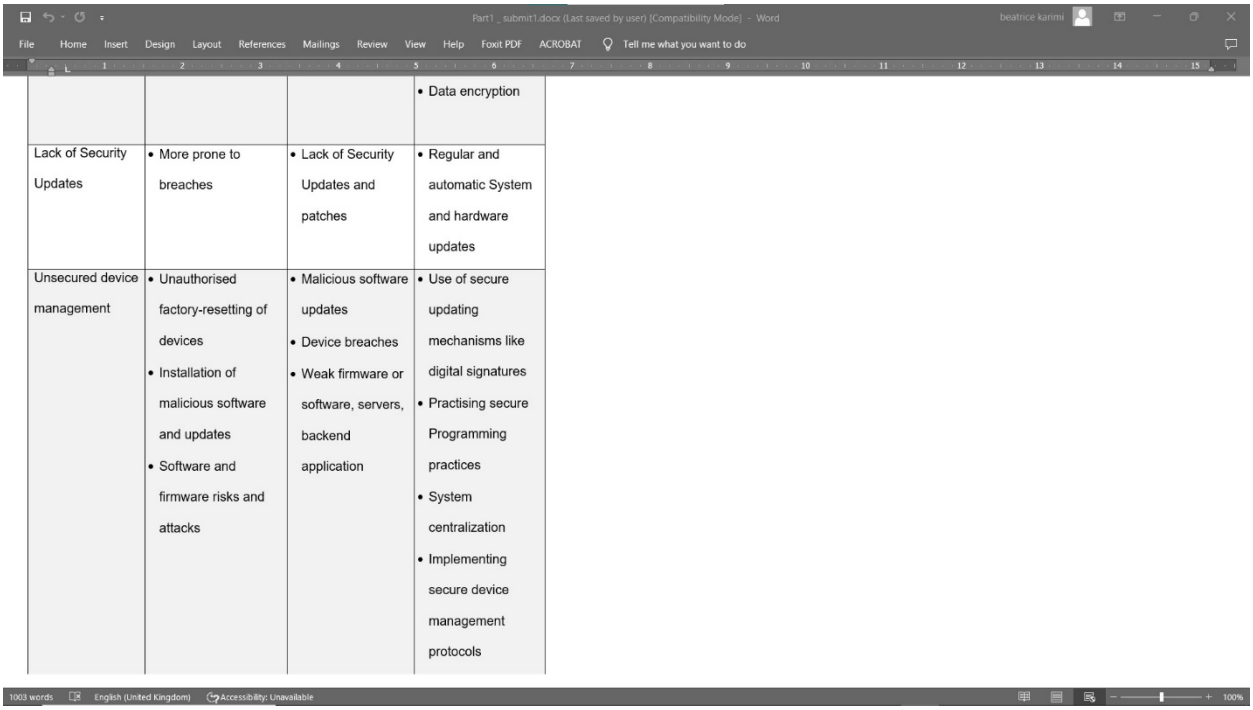
		27017, ISO 27018, etc	• Implement data privacy
The system's functionality is dependent on the Wi-Fi connection only,	• Wi-Fi dependency • Network attack • Denial-of-Service (DoS) and Denial-of-Sleep (DoSL) attacks	• System is down and security is compromised once Wi-Fi connection is lost or weak • Insecure network • Unencrypted communication	• Set-up other system connectivity e.g., Local Area Connection • Firewalls like Next-generation firewall • Limit device or network bandwidth • Backup connectivity options like 4G or 3G, to ensure that the system remains operational even if the Wi-Fi connection is lost.

Figure 6: My Contribution to 1st Part of the Group Assessment (D)



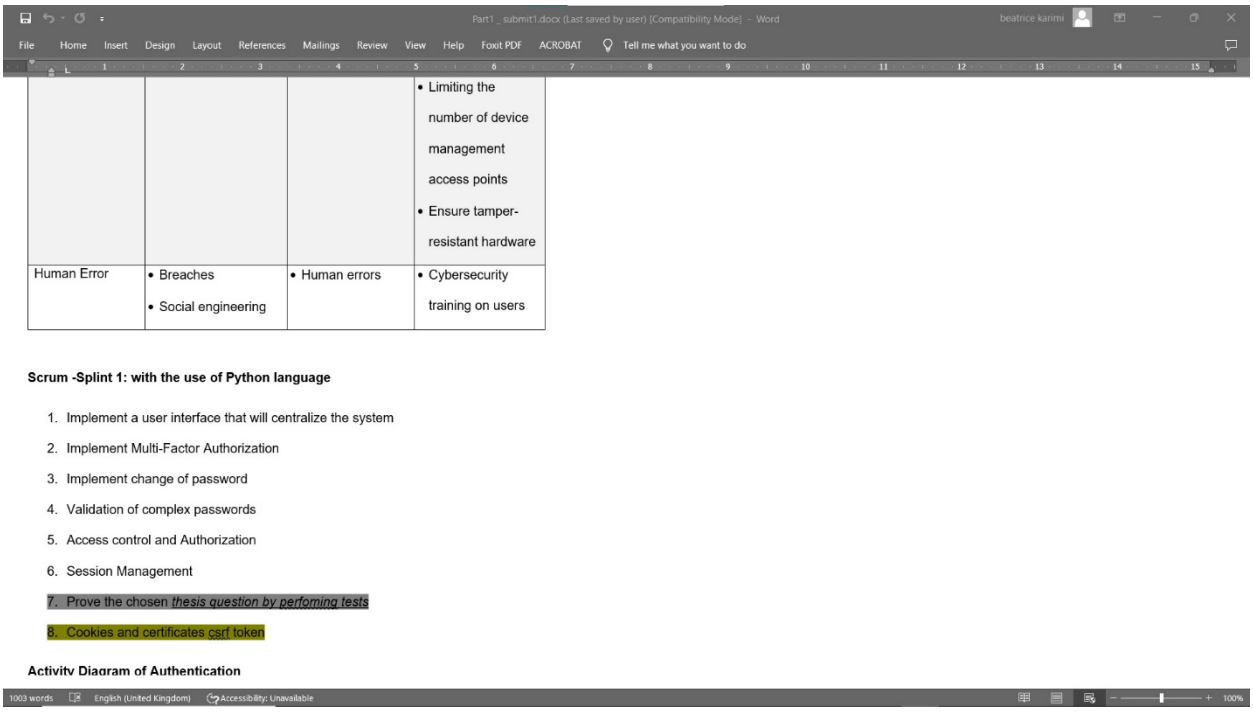
			• Intrusion Detection and Prevention Systems • Implementation of secure socket layer (SSL) Certificates, • Data Encryption • Network segmentation
Lack of security tests that make room for the system's improvements	• More prone to breaches	• Lack of security tests and scanning	• Regular security and backup testing, and scanning for threats helps in reinforcing the system
Lack of data storage security	• Injection attacks • Tampering	• Unsecure data storage	• Secure databases • Antivirus

Figure 7: My Contribution to 1st Part of the Group Assessment (E)



			• Data encryption
Lack of Security Updates	• More prone to breaches	• Lack of Security Updates and patches	• Regular and automatic System and hardware updates
Unsecured device management	• Unauthorised factory-resetting of devices • Installation of malicious software and updates • Software and firmware risks and attacks	• Malicious software updates • Device breaches • Weak firmware or software, servers, backend application	• Use of secure updating mechanisms like digital signatures • Practising secure Programming practices • System centralization • Implementing secure device management protocols

Figure 8: My Contribution to 1st Part of the Group Assessment (F)



			• Limiting the number of device management access points • Ensure tamper-resistant hardware
Human Error	• Breaches • Social engineering	• Human errors	• Cybersecurity training on users

Scrum -Splint 1: with the use of Python language

1. Implement a user interface that will centralize the system
2. Implement Multi-Factor Authorization
3. Implement change of password
4. Validation of complex passwords
5. Access control and Authorization
6. Session Management
7. Prove the chosen *thesis question by perfoming tests*
8. Cookies and certificates csrf token

Activity Diagram of Authentication

Figure 9: My Contribution to 1st Part of the Group Assessment (G)

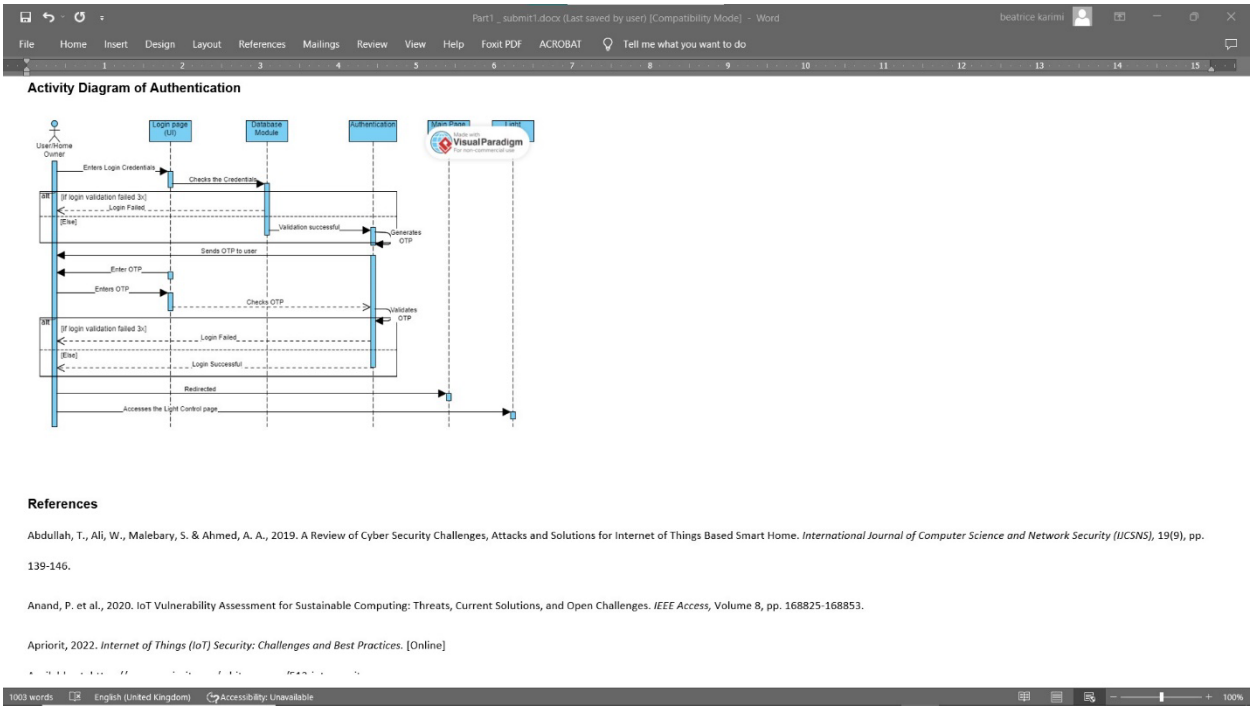


Figure 10: My Contribution to 1st Part of the Group Assessment (H)

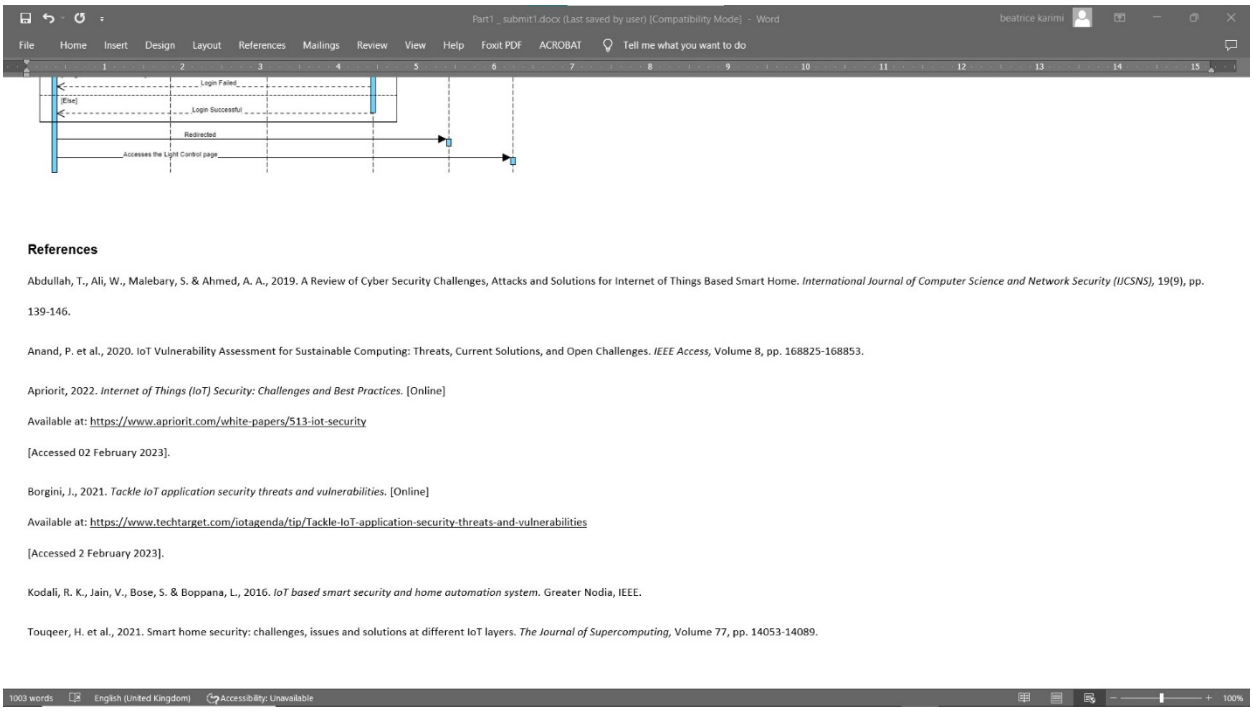


Figure 11: VS Code when Dockerizing the Django App

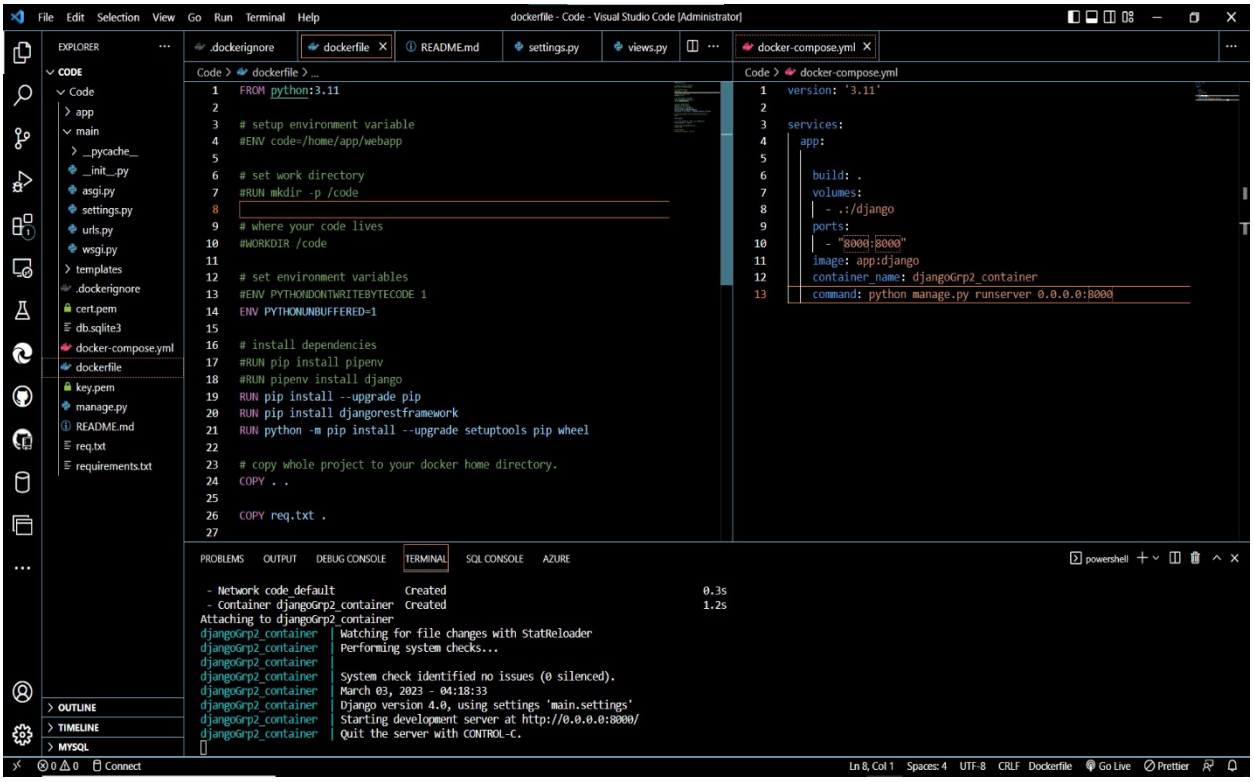


Figure 12: Docker Images of the Django App and its Snyk Test

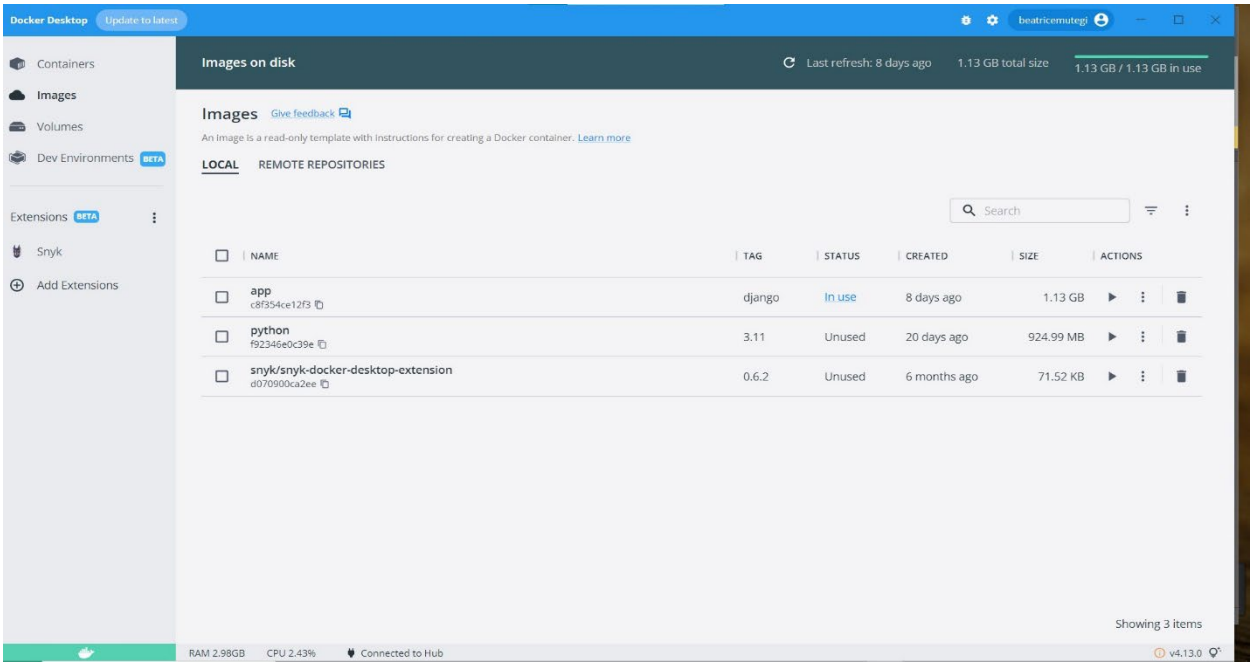


Figure 13:The Snyk Test Results of the Django App's Docker Image

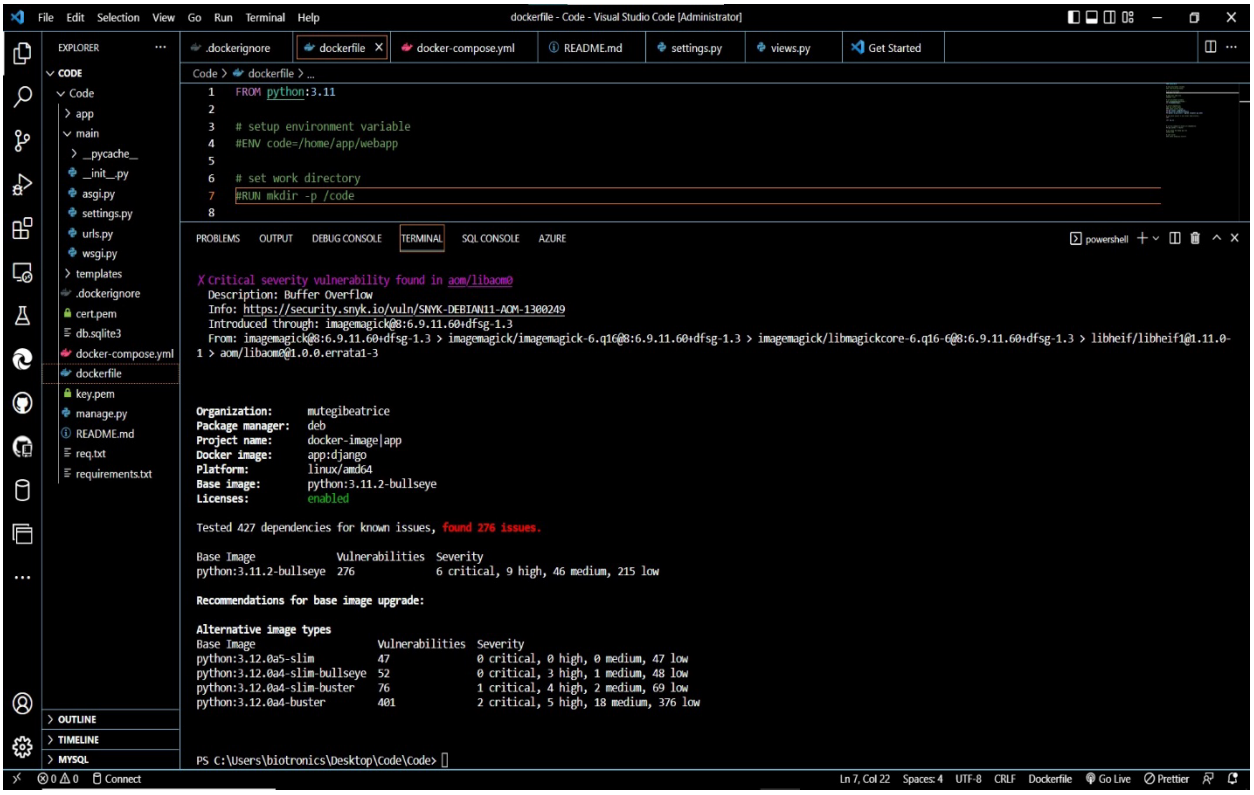


Figure 14: Docker Container of the Django App

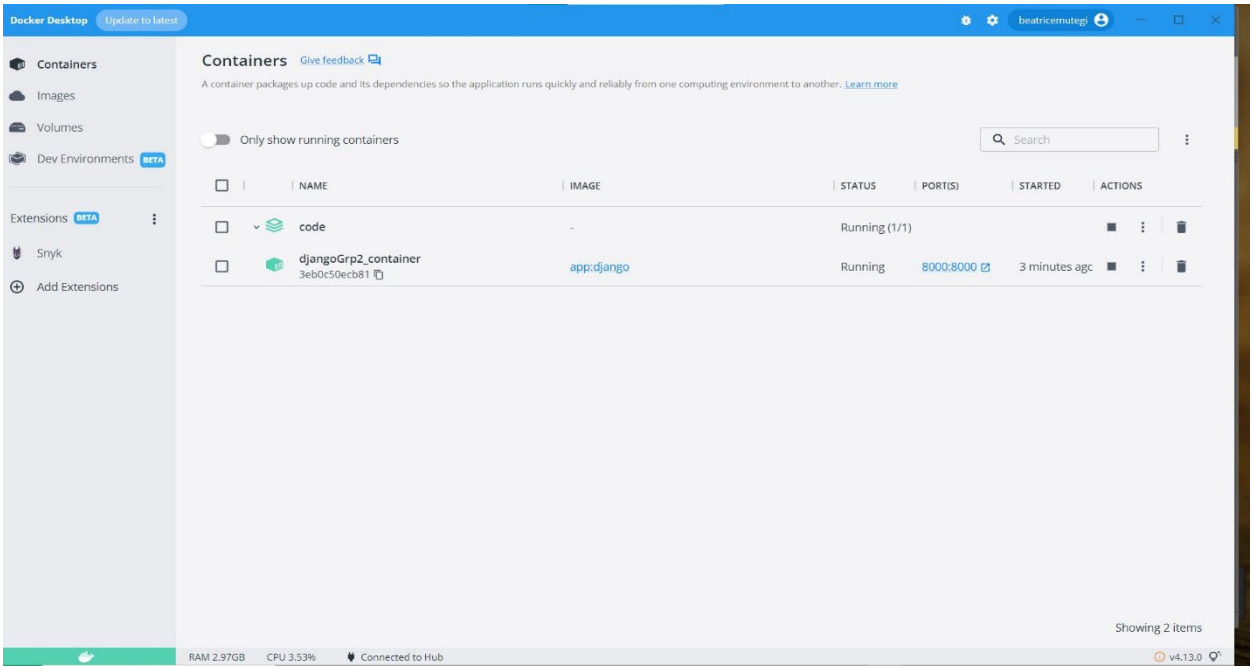


Figure 15: Running the Django App's Docker Container

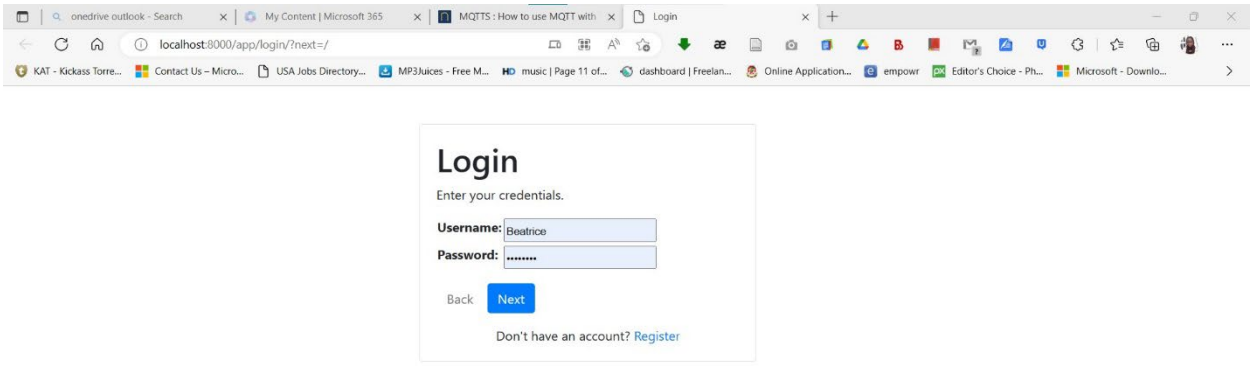


Figure 16: MQTT VS Code of the light control by motion folder.

Inclusive of the files: lightControl.py, motionDevice.py, running a pytest, mosquitto.config, password and the docker files.

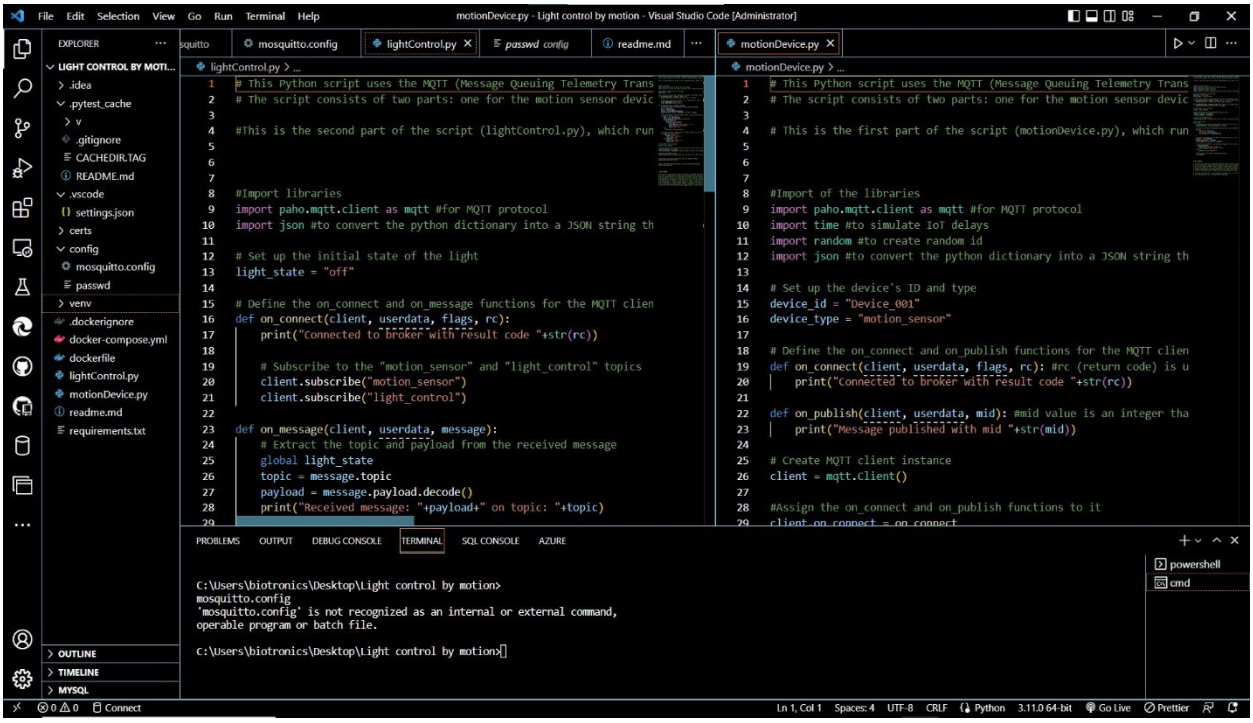


Figure 17: Running motionDevice.py script

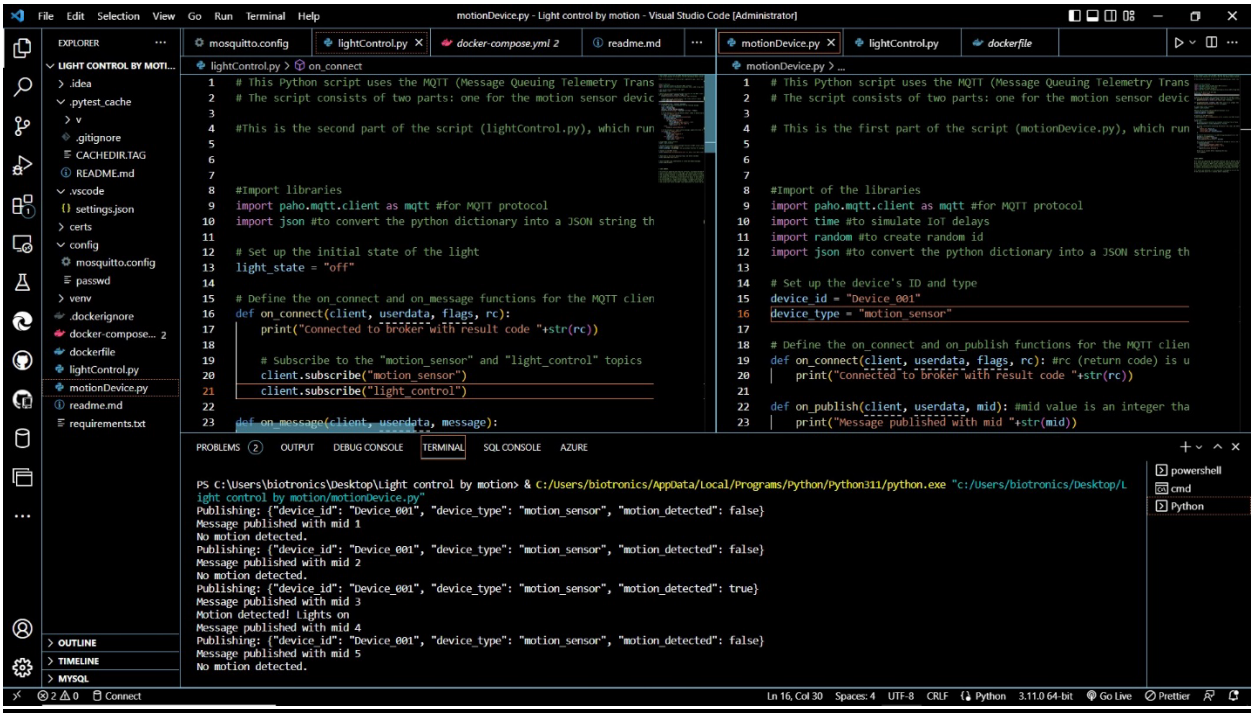


Figure 18: Running lightControl.py script