Next-Generation Threat Modelling!

Priya Joy Kaviyil

Atlantic Technological Universiy

Letterkenny, Ireland

L00171183@atu.ie

Danny McFadden

Atlantic Technological Universiy

Letterkenny, Ireland

Danny.McFadden@atu.ie

Ruth Lennon

Atlantic Technological Universiy

Letterkenny, Ireland

Ruth.Lennon@atu.ie

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—Threat Modelling, ChatGPT, Artificial Intelligence, Cybersecurity, security, risk assessment

# Introduction

Threat modelling is a crucial process in identifying, communicating, and managing security weaknesses[1]. Its main objective is to assess potential harm to an application or computer system by adopting the mindset of malicious hackers and identifying the types of threat agents that could cause damage. This is achieved through a comprehensive analysis of software architecture, business context, and other relevant artifacts such as functional specifications and user documentation[1]. Threat modelling provides a deeper understanding of critical aspects of the system and enables organizations to identify vulnerabilities and potential security threats. It is typically conducted during the design phase of a new application, although it can also occur at other stages of the software development lifecycle. During the threat modelling process, developers and security experts analyse the application's architecture, data flows, and access controls to identify potential threats and security vulnerabilities[1]. This helps developers to realize the security consequences of their design, code, and configuration choices, and to implement effective security measures to mitigate the identified threats.

To initiate the process of threat modelling, a diagram is often created that outlines the system's architecture, components, trust zones, and authentication flows [2] [3]. The inclusion of data flows in a diagram can prove to be extremely advantageous, as it provides a clear representation of how information is received and transmitted by the system, how it is altered, and where it is stored [2]. The primary goal of a data flow diagram (DFD) is to provide an overview of the system's scope and boundaries as a whole and a comprehensive analysis of the system's security posture.

There is a wide range of threat modelling methodologies that companies can make use of, as each is a unique approach and provides varied benefits. Among these, the most common are STRIDE, OCTAVE, TRIKE AND PASTA [4]. Threat modelling methodologies aids in generating a system abstraction and offering analyses of potential attackers, including their objectives and techniques. Moreover, it provides valuable insights on potential vulnerabilities and threats that may arise in the future. These are some of the best methodologies used, which have unique methods and frameworks to identify, analyse, measure, and sort threats [5].

This research paper proposes a novel methodology that differs from the conventional approaches to threat modelling outlined previously. It employs artificial intelligence (AI) to construct and evaluate threat models, specifically utilizing the AI chatbot ChatGPT. To threat model using this approach, a model is created which is a microservice connected to a MongoDB hosted on AWS cloud. The effectiveness of this technique in generating threat models is assessed in the research.

# Literature Review

Microsoft introduced the concept of threat modelling at the turn of the century[6]. It was formally documented in the book by Swiderski and Snyder [7] and was incorporated as a component in the initial release of the Microsoft Security Development Lifecycle (SDL) [8].

There are various existing approaches to threat modelling, varying from conceptual frameworks to practical tools. Myagmar, Lee, and Yurcik define threat modelling as a process aimed at comprehending the intricacies of a system and identifying all potential threats to it[9]. According to Shostack in [10], threat modelling typically involves two models: one representing the system to be built and another depicting the actual threats to the system and the mitigations. Similarly, in this paper, we are developing a system model that portrays the microservice connected to MongoDB and hosted in the AWS cloud, alongside a threat model that highlights the identified threats and the corresponding measures to mitigate them.

With regards to tools, Microsoft's Threat Modelling Tool (MS-TMT) [11] is a widely used freely available tool for threat modelling. There are other alternatives in the market, one such tool is OWASP Threat Dragon [12], which supports Windows, Linux and MacOS and a web app. While the drawing function in Threat Dragon may not be as user-friendly as desired, it does support Confidentiality, Integrity and availability(CIA) analysis and privacy threat modelling (LINDDUN) [6], and the STRIDE methodology [13]. Another tool available is SPARTA [14], which extends STRIDE threat modelling by using Data Flow Diagrams (DFDs) to link explicit countermeasures to each identified threat. Additionally, SPARTA includes simulations that estimate solution vulnerability, accounting for the capabilities of different types of attackers. For this paper, in the initial stage we are utilizing DFDs for the system model and the threat model to carry out the process of threat modelling the microservice system.

This paper aims to explore the effectiveness of using AI, specifically ChatGPT, to develop threat models, which differs from the traditional tools and methodologies mentioned above.

# Methodology

## Objective

Upon reviewing the available literature, several methodological approaches have been identified. that can be effectively employed for the purpose of conducting threat modelling. This paper aims to create a methodology to model cyber threats using ChatGPT which unlike the conventional methodologies which typically rely on manual processes that involve subject matter experts, stakeholders, and other key personnel to identify and assess potential threats to a system, creating a threat model using ChatGPT relies on an artificial intelligence language model that can analyse vast amounts of data and generate insights based on that analysis. The objective for this paper is creating a threat model for a microservice system connected to a MongoDB instance and hosted in AWS using ChatGPT which would assist in identifying potential threats and vulnerabilities that could impact the system's security. And by analysing the system and its associated risks, developing a comprehensive threat model that outlines specific measures to mitigate the identified threats.

## Approach

This section presents the methodology used in the paper to establish a structured threat modelling approach for a microservice system. Firstly, a Data Flow Diagram (DFD) is created for the system model, which outlines the various dataflows and services used in the system. The system model serves as the input for asking ChatGPT to create the threat model, which is then used to generate a threat model for the system. Based on the outputs provided by ChatGPT, a DFD for the threat model is created, which outlines the potential threats that could impact the system's security. To address these threats, solutions are asked to ChatGPT which again will be added to the threat model DFD and implemented to mitigate the identified threats.

### System Model: A model is created for a spring boot microservice built which is connected to a mongo dB database (Fig.1). Both the microservice and database are hosted in AWS EC2 virtual machine using the EC2 and S3 bucket AWS services. Created an AWS Linux EC2 instance from the AWS Management Console by launching a new instance and configured its settings, including selecting the desired Linux AMI. Security groups were set up to allow incoming connections on MongoDB and microservice ports. Accessed the EC2 instance from local terminal using a secret key pair by setting the key file's permissions and using SSH. Downloaded and installed MongoDB on the EC2 instance, updated its configuration to accept remote connections, and created a MongoDB user. Installed MongoDB Compass on the local machine and connected to the MongoDB instance on the EC2 instance. Created a Spring Boot microservice with IntelliJ (Appendix B), built and compiled it to generate a JAR file, and created an S3 bucket in AWS to store the JAR. Jar file was uploaded to the S3 bucket and copied to the EC2 instance. Successfully started the microservice application on the EC2 instance and tested its endpoints using a tool like Postman, specifying the EC2 instance's public IP address or DNS name (Appendix A).

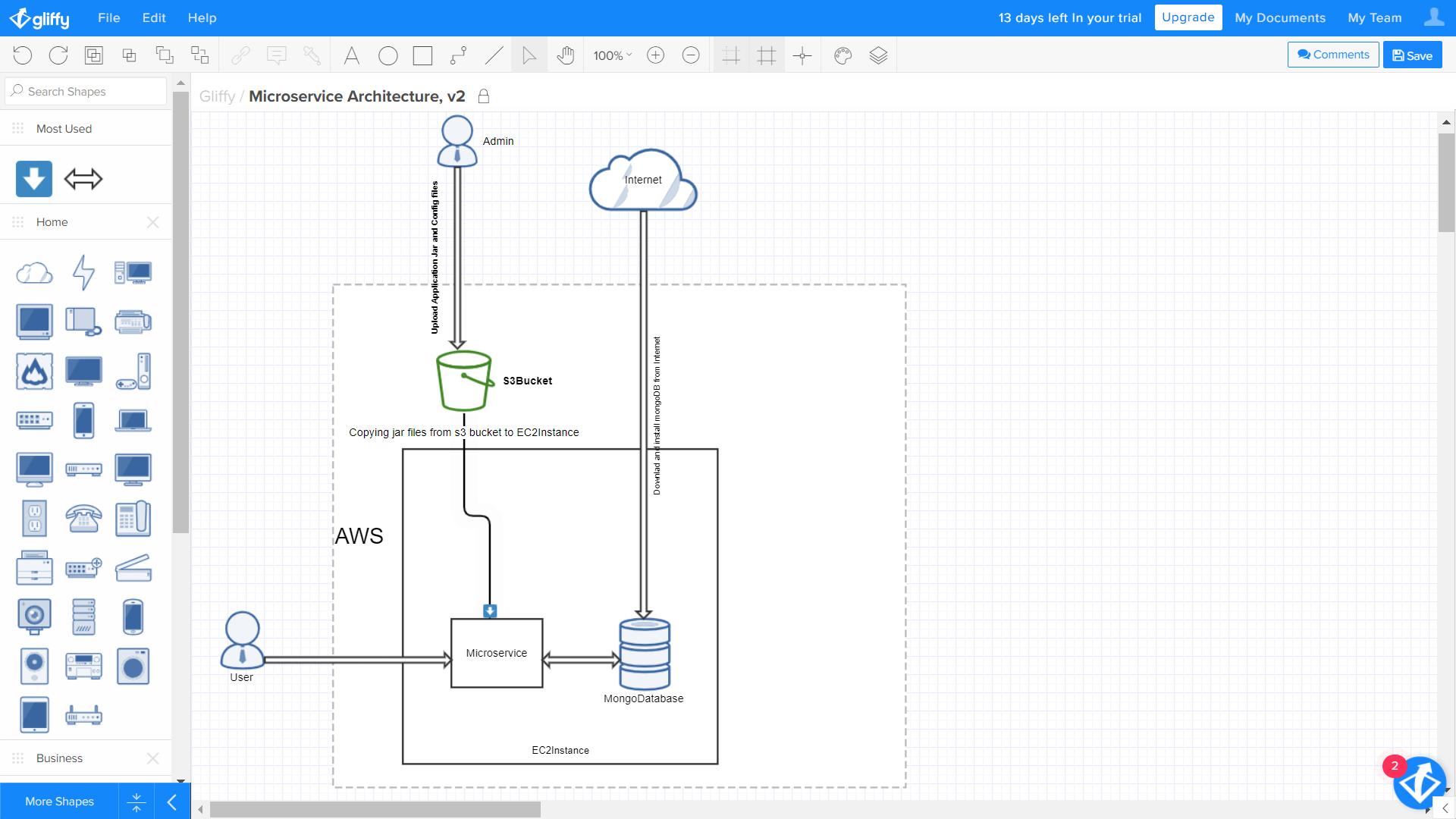


Fig.: Microservice System Model

### Threat Model: Asked ChatGPT to create a threat model for the microservice system model created. ChatGPT provided a list of potential threats (Fig.2). A threat model has been developed solely based on the provided information (Fig.3). According to ChatGPT, here are the potential threats to consider for the system:

* Unauthorized access to EC2 instance and MongoDB database.
* Injection attacks, such as SQL, NoSQL, or command injection.
* Insecure communication channels, leading to interception or manipulation of data.
* Insufficient authentication and authorization mechanisms.
* Denial of service (DoS) attacks, causing unavailability or unresponsiveness.
* Weak passwords or credentials.

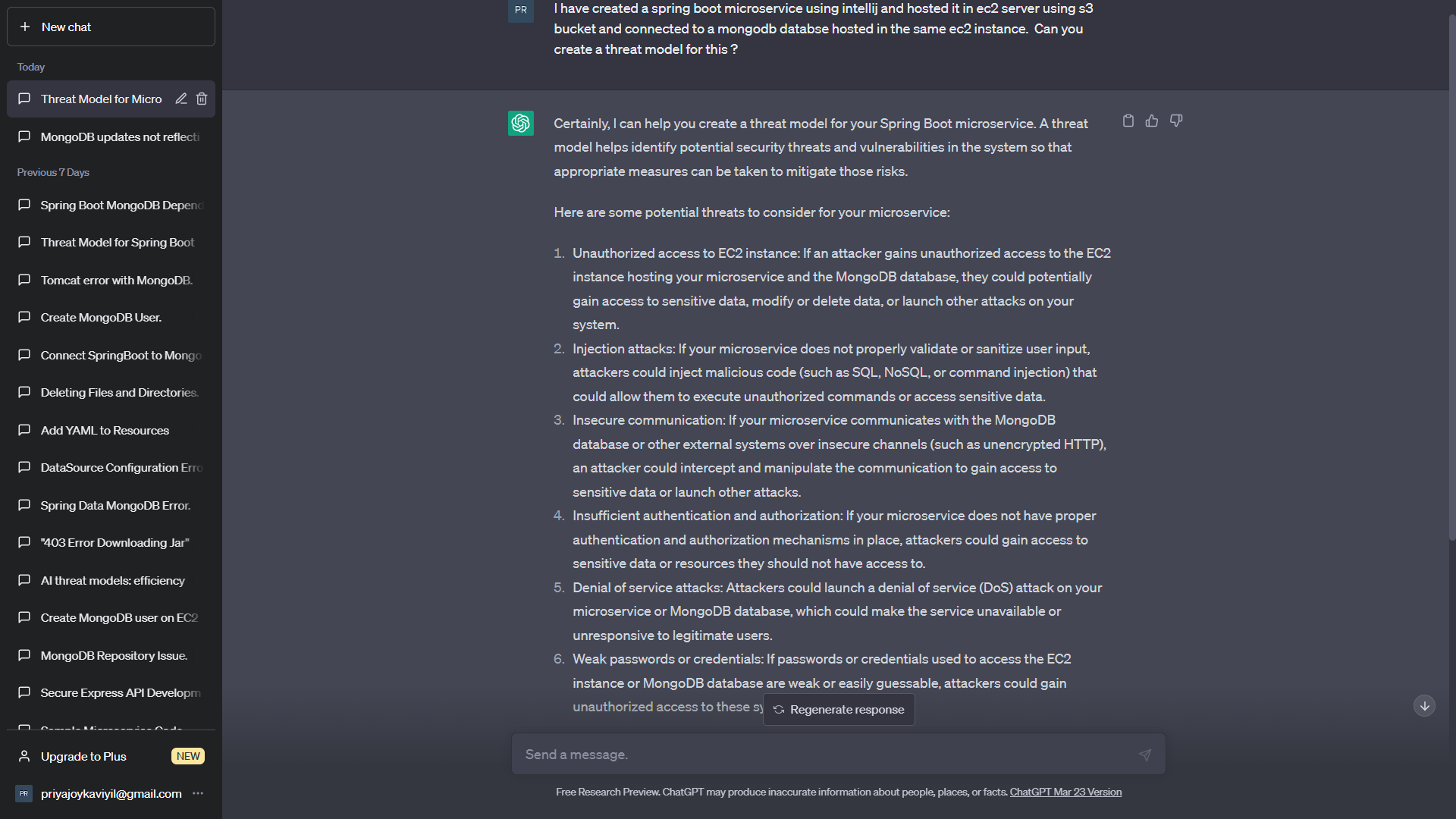


Fig.2: Asking ChatGPT to create threat model

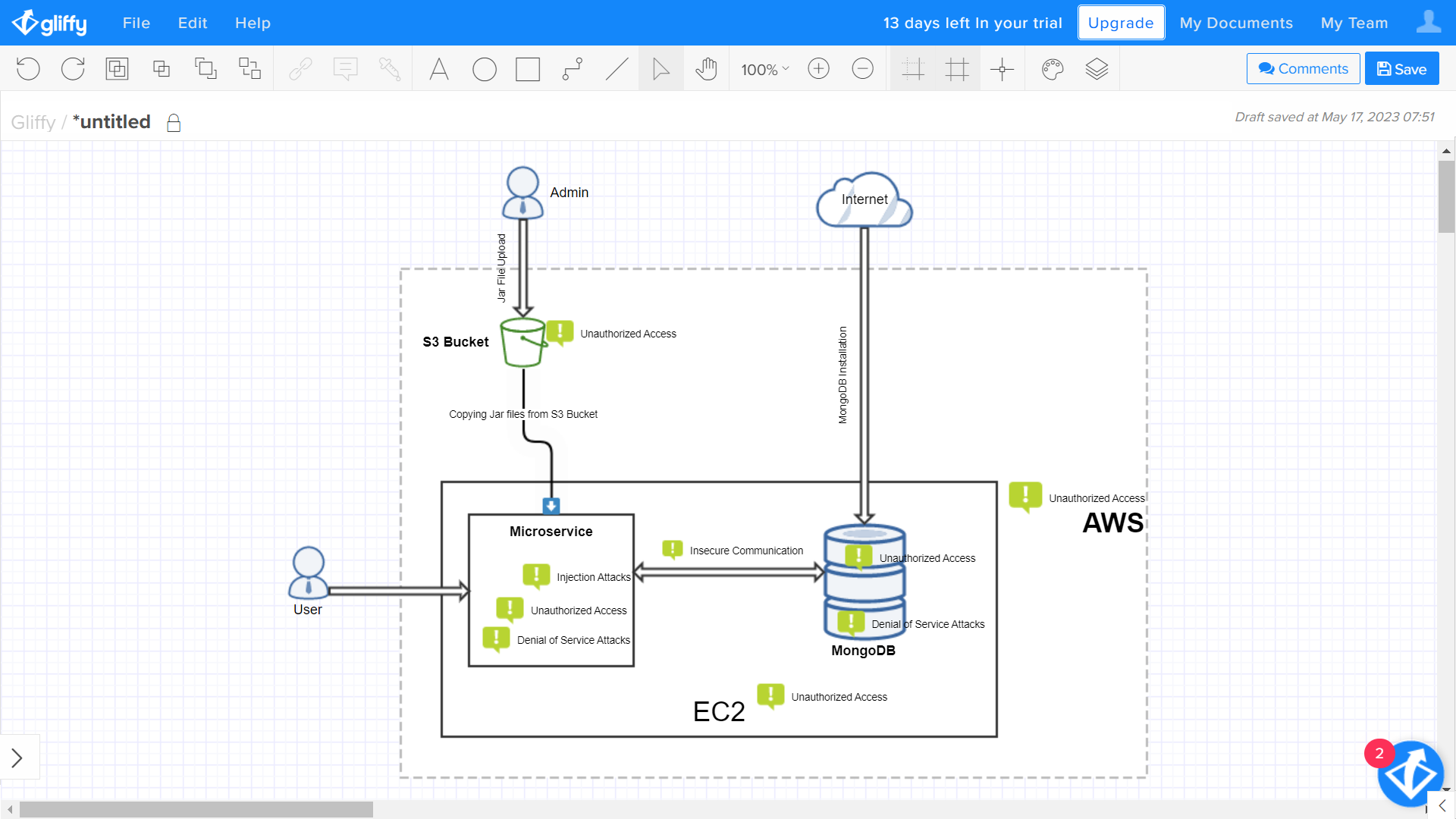


Fig.3: ChatGPT Created Threat Model

### Threat Mitigations: Corresponding mitigation measures for the potential threats were provided by ChatGPT (Figure 4, Figure 5, Appnedix A). According to ChatGPT here are the key points for ensuring the security of your Spring Boot microservice, EC2 instance, and MongoDB database:

For the EC2 instance and MongoDB database:

* Use strong and unique passwords.
* Regularly apply security updates and patches.
* Configure access controls and firewall rules.
* Implement encryption for data at rest and in transit.
* Set up monitoring and logging.
* Take regular backups of the database

For preventing injection attacks

* Validate input data to ensure the expected format and data types.
* Use prepared statements or parameterized queries.
* Sanitize user input by removing or encoding unnecessary characters.
* Utilize security-focused libraries for authentication and authorization.
* Implement input filtering and length limitations

For implementing authentication and authorization mechanisms:

* Use multi-factor authentication, such as MFA.
* Implement role-based access control (RBAC).
* Enforce strong password policies and secure session management.
* Implement API gateway security to control access

For DoS protection:

* Implement rate limiting and traffic filtering.
* Utilize load balancing and caching mechanisms.
* Consider using cloud-based DoS protection services.

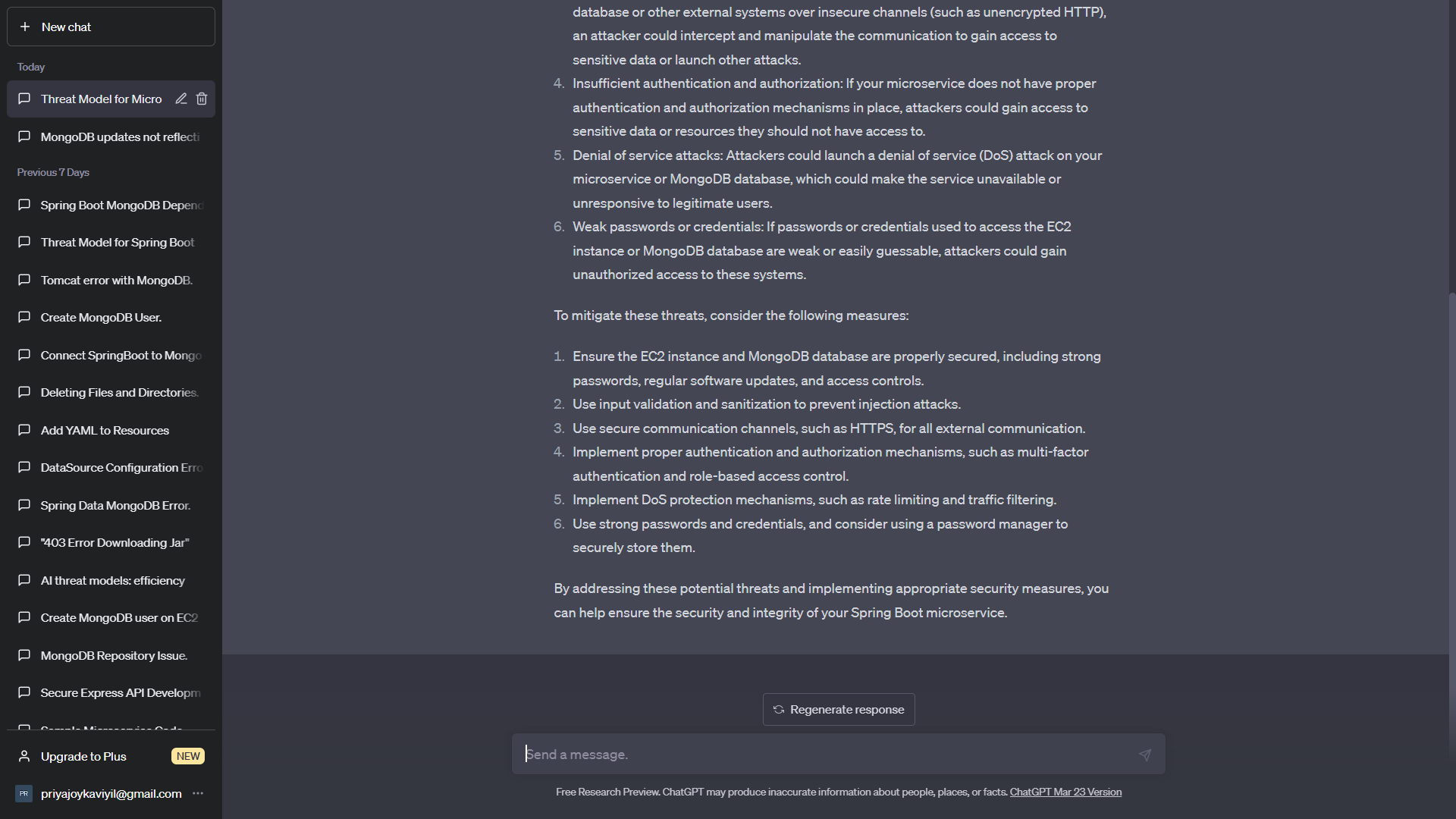


Fig.4: Mitigations

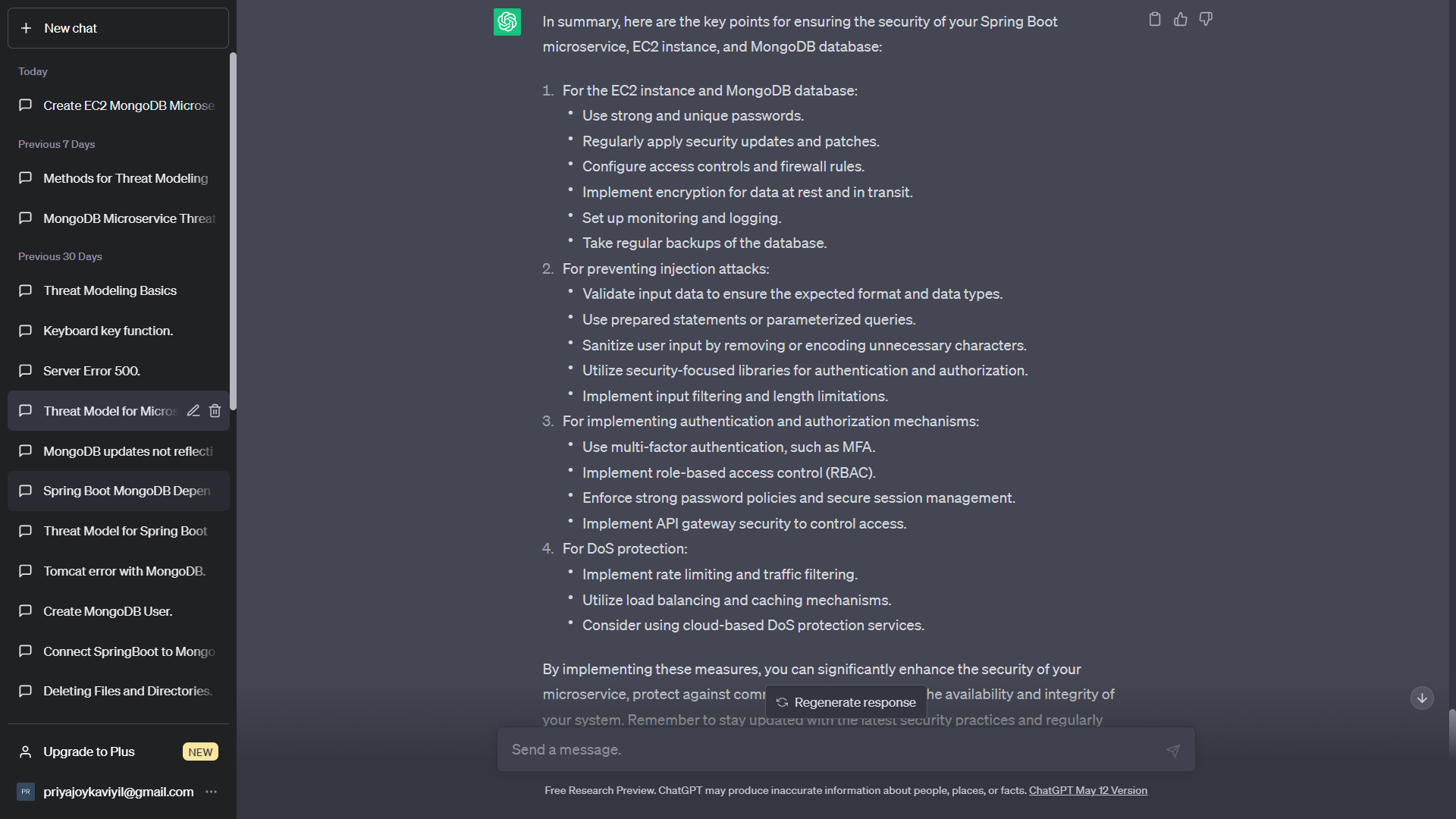


Fig.5 Mitigation Measures

### Implementation: Created a mitigation model from the mitigation and measures information taken from ChatGPT (Fig.6).

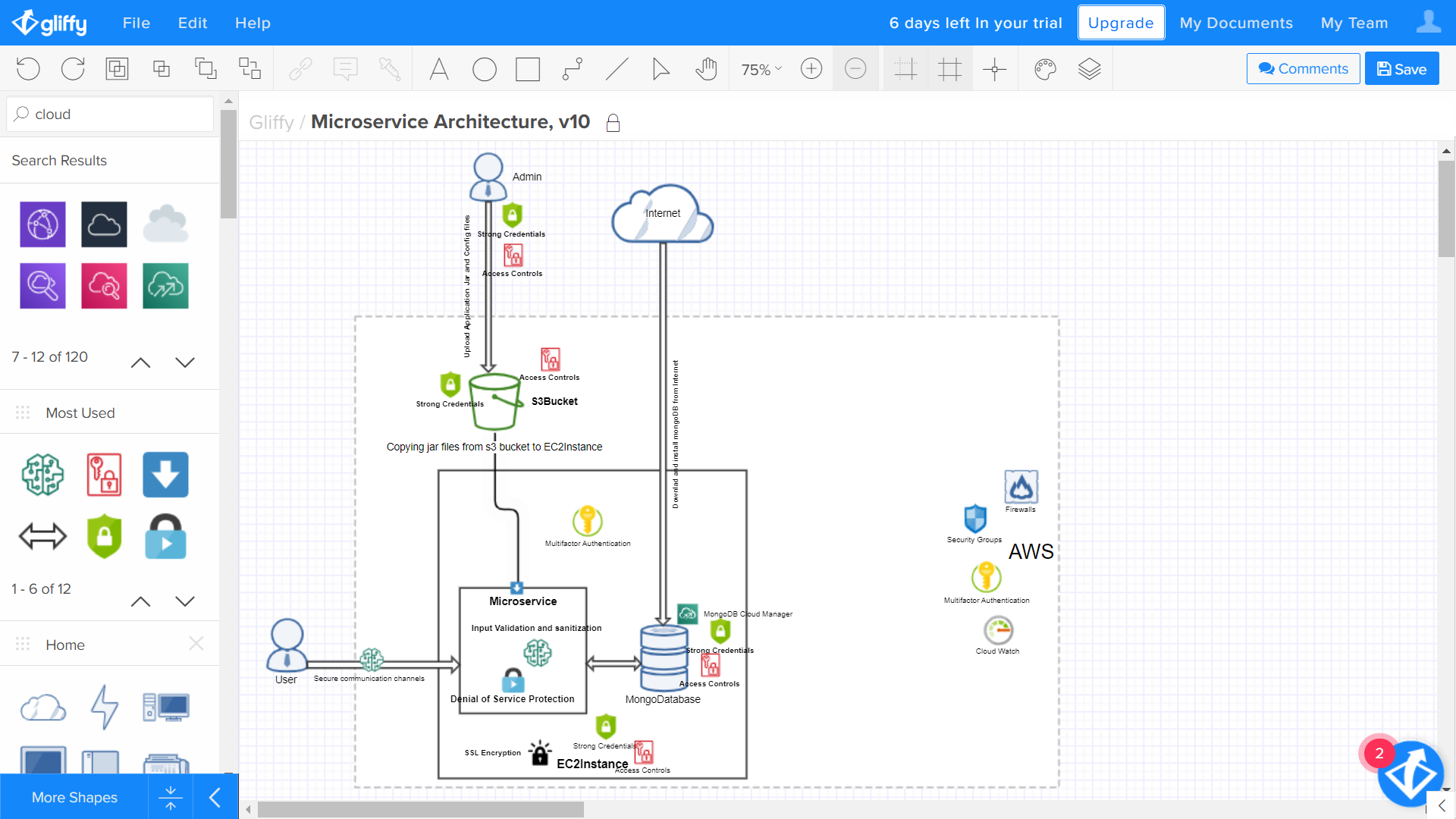


Fig.6: Implementing ChatGPT provided Mitigations

# Case Study

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
* Use a zero before decimal points: “0.25”, not “.25”. Use “cm3”, not “cc”. (*bullet list*)

## Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Conclusion

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Appendice A

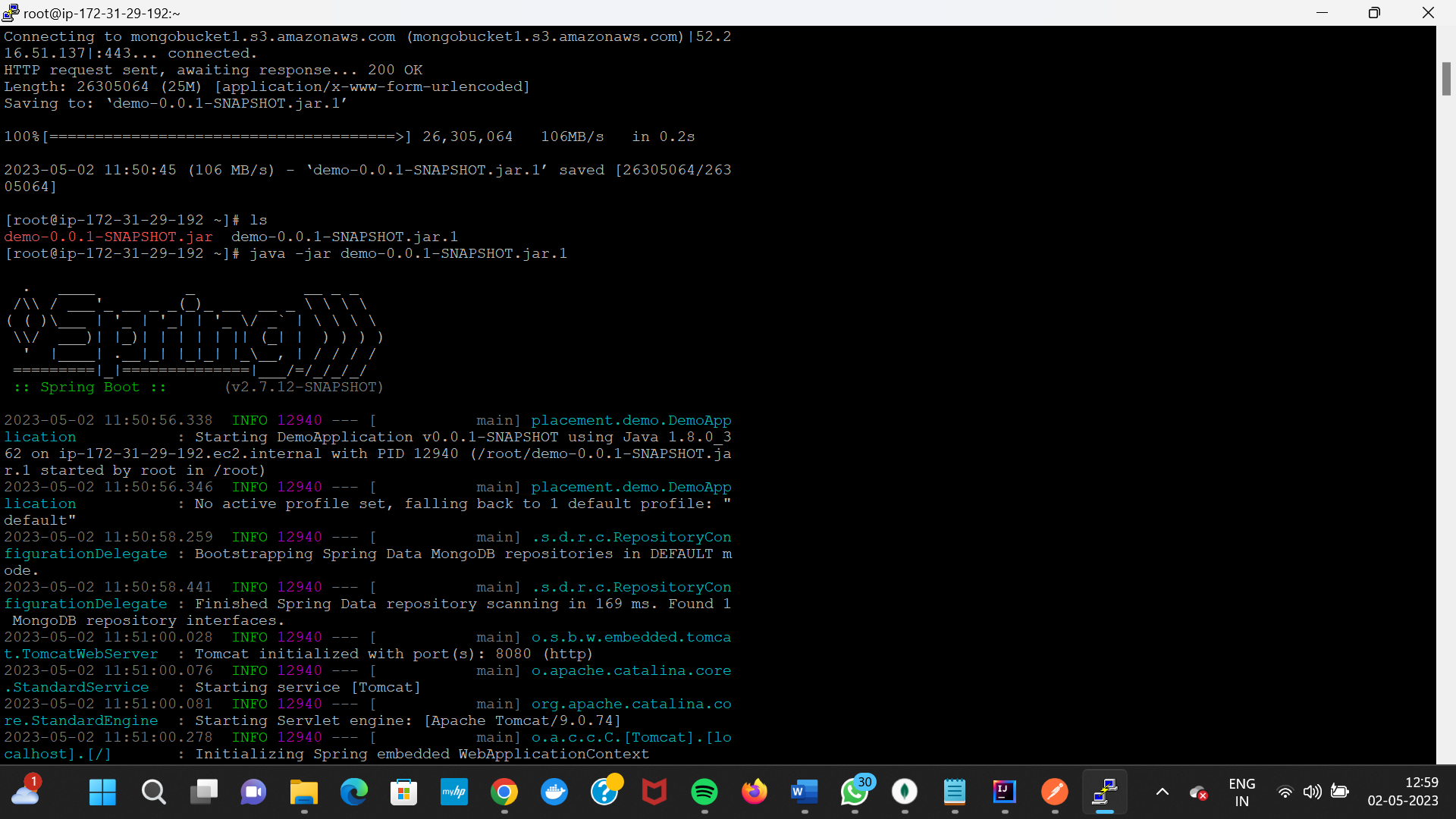


Figure : Microservice Application Running in EC2 Server -A

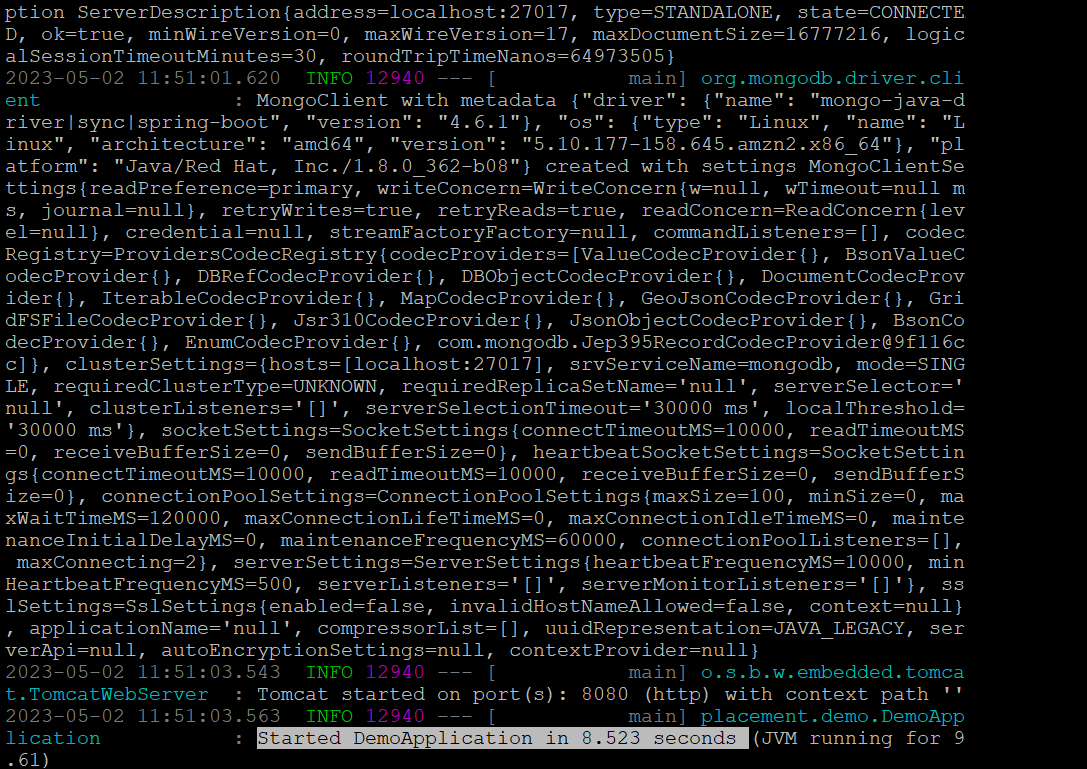


Figure : Microservice Application Running in EC2 Server -B

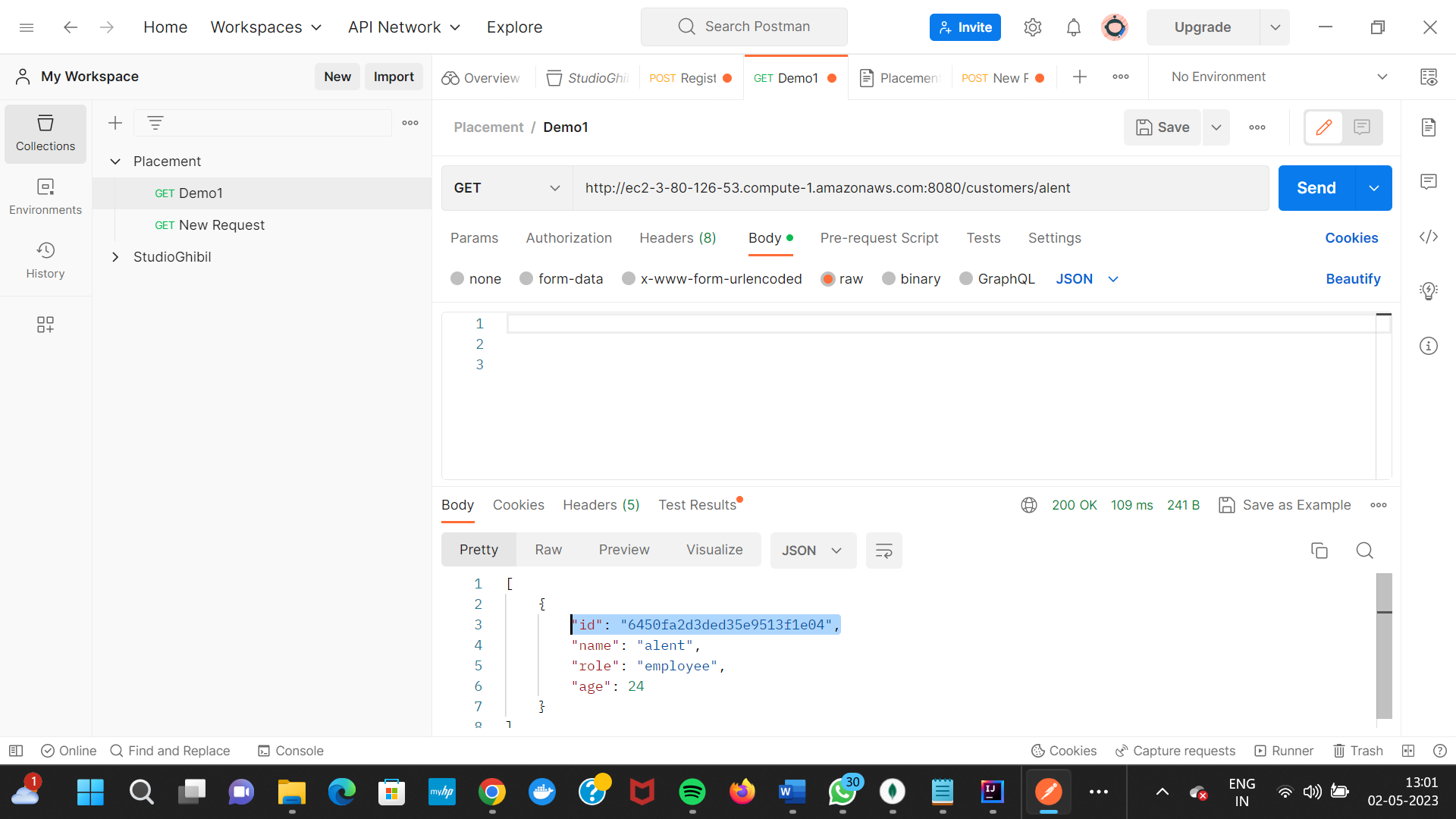


Figure : Testing the endpoints in Postman

##### Appendix B

GitHub Repo of the microservice application created: <https://github.com/L00171183/ThreatModel/tree/main/demo>

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

[1] “Threat Modeling - What, Why & How (eBook).pdf.”

[2] “What Is Threat Modeling and How Does It Work? | Synopsys.” https://www.synopsys.com/glossary/what-is-threat-modeling.html (accessed May 10, 2023).

[3] “Threat Modeling Process | OWASP Foundation.” https://owasp.org/www-community/Threat\_Modeling\_Process (accessed May 10, 2023).

[4] “Methodologies Blog,” Jan. 23, 2023. https://www.iriusrisk.com/resources-blog/methodologies (accessed May 10, 2023).

[5] M. T, “Top 5 Threat Modeling Methodologies,” Practical DevSecOps, Feb. 06, 2023. https://www.practical-devsecops.com/threat-modeling-methodologies/ (accessed May 10, 2023).

[6] K. Bernsmed, D. S. Cruzes, M. G. Jaatun, and M. Iovan, “Adopting threat modelling in agile software development projects,” J. Syst. Softw., vol. 183, p. 111090, Jan. 2022, doi: 10.1016/j.jss.2021.111090.

[7] F. Swiderski and W. Snyder, Threat Modelling. [Online]. Available: http://refhub.elsevier.com/S0164-1212(21)00187-4/sb32

[8] M. Howard and S. Lipner, The Security Development Lifecycle. [Online]. Available: http://refhub.elsevier.com/S0164-1212(21)00187-4/sb13

[9] A. J. Lee, G. A. Koenig, and W. Yurcik, “Cluster Security with NVisionCC: The Forseti Distributed File Integrity Checker”.

[10] A. Shostack, Threat modeling: Designing for security. [Online]. Available: http://refhub.elsevier.com/S0164-1212(21)00187-4/sb28

[11] “Microsoft Security Development Lifecycle Threat Modelling.” https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling (accessed May 11, 2023).

[12] “OWASP Threat Dragon.” https://www.threatdragon.com/#/ (accessed May 11, 2023).

[13] E. Bygdas, L. A. Jaatun, S. B. Antonsen, A. Ringen, and E. Eiring, “Evaluating Threat Modeling Tools: Microsoft TMT versus OWASP Threat Dragon,” in 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Dublin, Ireland: IEEE, Jun. 2021, pp. 1–7. doi: 10.1109/CyberSA52016.2021.9478215.

[14] L. Sion, D. Van Landuyt, K. Yskout, and W. Joosen, “SPARTA: Security & Privacy Architecture Through Risk-Driven Threat Assessment,” in 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), Seattle, WA: IEEE, Apr. 2018, pp. 89–92. doi: 10.1109/ICSA-C.2018.00032.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**