INFO1111: Computing 1A Professionalism

2025 Semester 1

Skills: Team Project Report

Submission number: T4 SL Group 6 (1)

Github link: `https://github.com/KaltsitFan/INFO1111_GROUP.git`

Team Members:

Name	Student ID	Target * Foundation	Target * Advanced	Selected Major
Fan Kaffa	510041359	AG	NA	Computer Science
Lin Link	540801400	A	NA	Data Science
Song Jared	550155230	A	NA	SW Development
Zheng Barnett	550718806	A	NA	Cyber Security

* Use the following codes:

- NA = Not attempting in this submission
- A = Attempting (not previously attempting)
- AW = Attempting (achieved weak in a previous submission)
- AG = Attempting (achieved good in a previous submission)
- S = Already achieved strong in a previous submission

Contents

1. Group Response 2
2. Individual Response 3
   - 2.1. Skills for Computer Science: Kaffa Fan 3
   - 2.2. Advanced Submission Kaffa (Computer Science) 4
   - 2.3. Skills for SW Development: Jared Song 10
   - 2.4. Skills for Cybersecurity: Barnett Zheng 13
3. Submission contribution overview 21
   - 3.1. Submission 1 contribution overview 21
   - 3.2. Submission 2 contribution overview 21
   - 3.3. Submission 3 contribution overview 21

1. Group Response

Our team, composed of members from different computing majors, effectively utilized GitHub's issue tracker for task allocation, enabling clear responsibility assignments and progress tracking. Cross-disciplinary collaboration played a crucial role—our Software Development member assisted the Cybersecurity member in coding challenges, while our Data Science member guided data-related tasks. Initially, LaTeX documentation presented significant challenges due to frequent syntax errors and lack of real-time previews. Transitioning to Visual Studio Code with LaTeX extensions resolved these issues by enabling immediate error detection and correction. Through structured GitHub management and mutual technical support, our collaborative approach markedly improved communication, efficiency, and overall project management effectiveness.

2. Individual Response

2.1. Skills for Computer Science: Kaffa Fan

In the disaster response system project, my primary responsibilities included the establishment of the overall system architecture, the setup of the GitHub collaboration framework, and the configuration of the LaTeX documentation environment. Reflecting based on the SFIA framework, I selected the following two skills:

Software Development (PROG)

According to the SFIA framework [1], software development is foundational to implementing technological solutions. For the LA wildfire disaster response scenario, effectively managing documentation using LaTeX was critical to ensure rapid generation and dissemination of accurate reports. Initially, our team frequently encountered LaTeX syntax errors such as missing commands missing commands (\enditemize) due to using editors without real-time preview capabilities. By transitioning to Visual Studio Code with LaTeX extensions, we could immediately identify and rectify these errors, significantly reducing documentation delays. This directly supported emergency response requirements, enabling quick delivery of clear instructions and updates to responders. Additionally, my demonstrated ability to compile and manage documents via the Git terminal further ensured smooth, efficient project documentation workflows.

Skill Enhancement (Team Collaboration Details): Introducing real-time preview tools considerably enhanced the team's productivity and accuracy, minimizing errors and speeding up response times—key factors during an emergency.

Areas for Further Improvement: Despite improvements, I identified an over-reliance on tool-based error detection. In future projects, I aim to improve my manual code review skills, ensuring reliability even when technological resources are limited, a critical capability in disaster situations.

Systems Integration and Build (SINT)

According to the SFIA framework [1], effective systems integration is essential for creating a coherent and responsive disaster management system. In the context of the wildfire scenario, my role involved integrating various system components, such as real-time data interfaces and a user-friendly incident dashboard for emergency personnel. Early in the project, unclear task assignments created confusion among team members. Implementing GitHub Issues significantly clarified roles and responsibilities, specifically tasks like "Frontend Interface (Incident Dashboard)" and "Data Interface Development (Real-time Sensor Feeds)," streamlining integration processes. My consistent use of Git for task management and module synchronization (Figures 4) ensured efficient module integration, crucial for rapid deployment in emergency scenarios.

Skill Enhancement (Team Collaboration Details): Clear task definition through GitHub substantially improved our integration capability, facilitating quick and smooth development of crucial disaster-response features, such as real-time dashboards.

Areas for Further Improvement: Our technical discussions sometimes involved excessive jargon, complicating interdisciplinary communication. Moving forward, simplifying complex technical concepts into clear, accessible language will be essential, particularly in high-stakes, collaborative disaster-response environments.

2.2. Advanced Submission Kaffa (Computer Science)

Project Management and Progress Overview

Throughout the project, I established and utilized GitHub Projects along with Issue tracking functionalities to coordinate tasks, assign responsibilities, and monitor overall project progress effectively. The built-in project chart feature provided by GitHub enabled team members to visually track task completion rates, facilitating a clear understanding of the project's progression and enhancing our overall productivity.

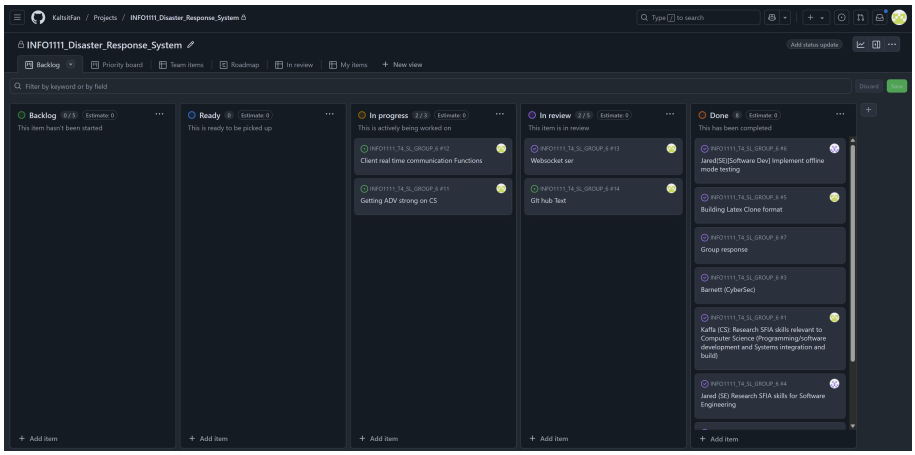


Figure 1: GitHub project board displaying Issues and their statuses

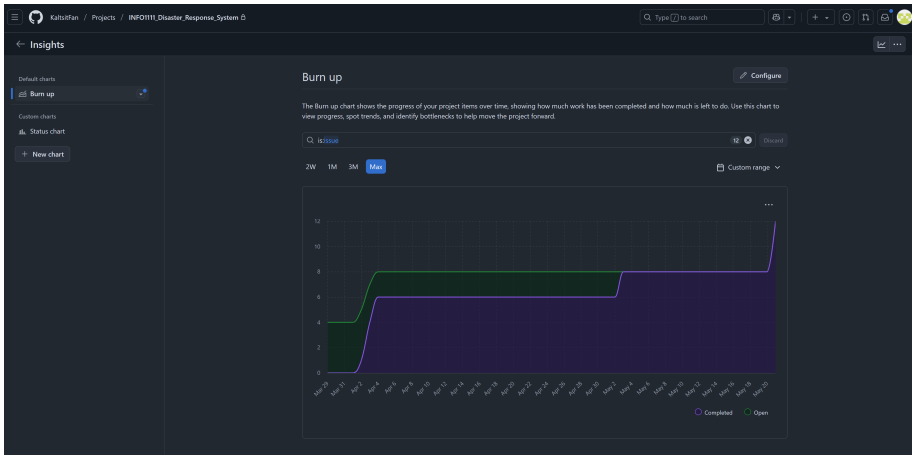


Figure 2: GitHub project progress line chart

Tool Exploration: Python WebSockets (Part A)

Main Functionalities of Python WebSockets Python WebSockets offers robust support for real-time bidirectional communication between clients and servers. Its primary functionalities include:

- **Full Duplex Communication:** Allows simultaneous sending and receiving of data.
- **Event-Driven Data Push:** Enables servers to proactively push updates to clients, eliminating continuous polling and thus ideal for real-time applications.

- **Seamless Integration and Support:** Python's websockets library integrates effortlessly with the asyncio framework, simplifying asynchronous programming and client-server interactions.

Importance in Computer Science WebSockets play a pivotal role in applications that require real-time interaction, such as online collaboration tools, multiplayer games, and real-time monitoring dashboards. Specifically, in disaster response scenarios, WebSockets significantly improve system responsiveness by providing instantaneous data updates, enabling emergency responders to quickly react and mitigate risks efficiently.

Limitations of WebSockets Despite its significant advantages, WebSockets has certain limitations:

- **Limited Efficiency for Media Streams:** WebSockets is less efficient for streaming audio and video.
- **Complexity in Managing Connection Reliability:** Developers need to manually handle reconnection logic after interruptions, increasing implementation complexity.
- **Resource Intensive:** Continuous connections consume substantial server resources, especially in large-scale systems, requiring careful infrastructure planning and optimization.

Technical Implementation and Reflective Learning (Part B)

Technical Implementation: Basic WebSockets Example I developed a basic WebSockets demonstration showcasing client-server interaction. The application successfully establishes a WebSocket connection enabling real-time messaging, effectively illustrating core WebSocket functionality and typical application scenarios.

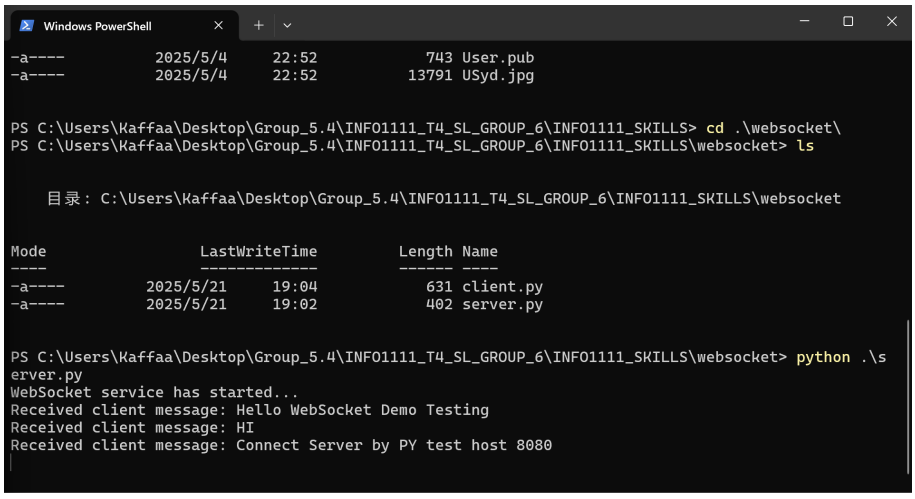


Figure 3: Terminal output demonstrating server startup and client connections

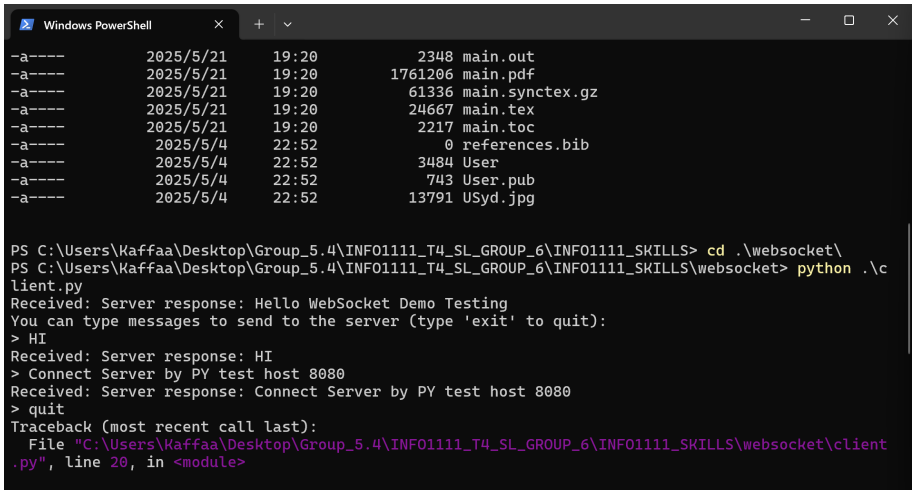


Figure 4: Example of message exchanges between client and server

Reflective Learning Initially, I encountered challenges related to environment setup and mastering asynchronous data handling, particularly regarding issues such as port mismatches and conflicts causing connection failures. To resolve these problems, I actively engaged with online forums like Stack Overflow, consulted official documentation, and conducted iterative testing with my team. Through this practical exercise, I not only gained deeper insights into WebSockets technology but also understood its strategic value in building highly responsive and real-time systems, especially applicable to critical scenarios like disaster management [2] [3].

3. Git Response

1. Clone the repository

Clone the remote repository to your local computer:

```
git clone https://github.com/KaltsitFan/INFO1111_T4_SL_GROUP_6
```

This picture illustrates the cloning process:

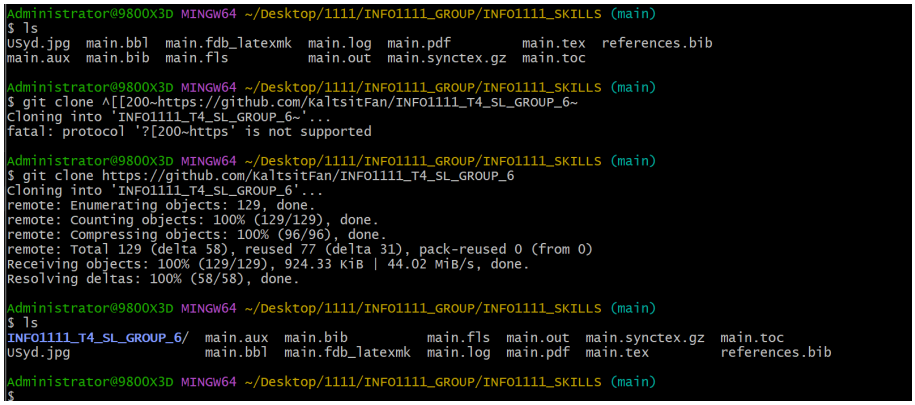


Figure 5: Cloning the repository

2. Stage, Commit, Push, and Check Status

Stage all modified files, commit the changes with a clear message, push the committed changes to GitHub, and verify the repository status:

```
git add .
git commit -m "Your commit message"
git push
git status
```

This picture demonstrates staging, committing, pushing, and checking status:

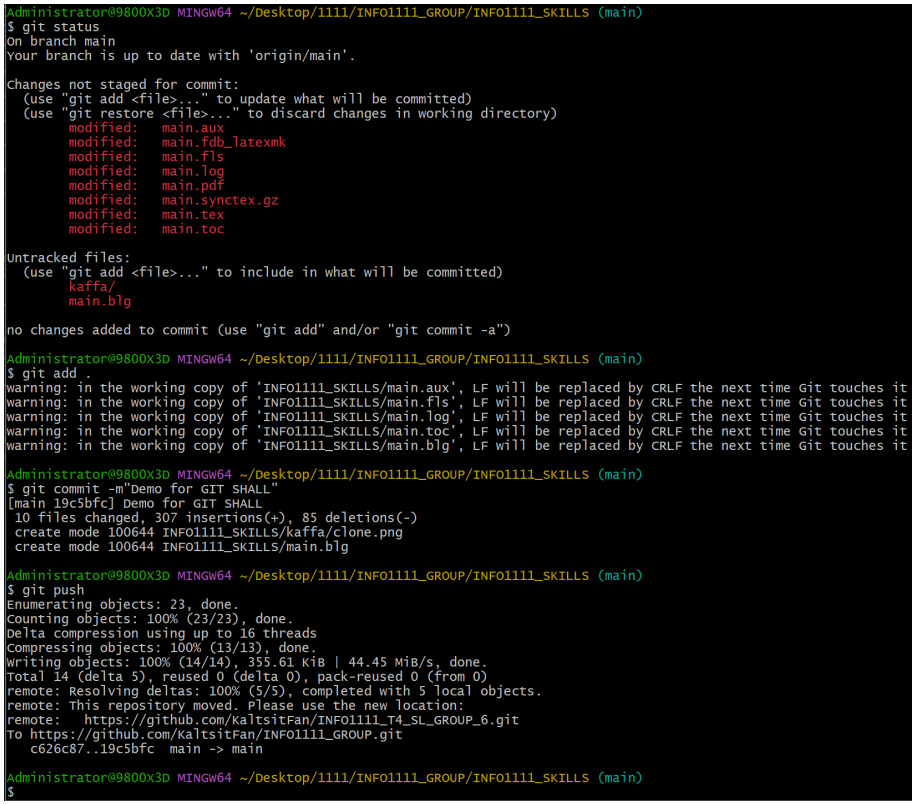


Figure 6: Staging, committing, pushing changes, and checking status

3. Synchronize with remote repository

Pull the latest updates from the GitHub repository to synchronize your local repository:

```
git pull origin main
```

This picture shows synchronizing the local repository with remote updates:

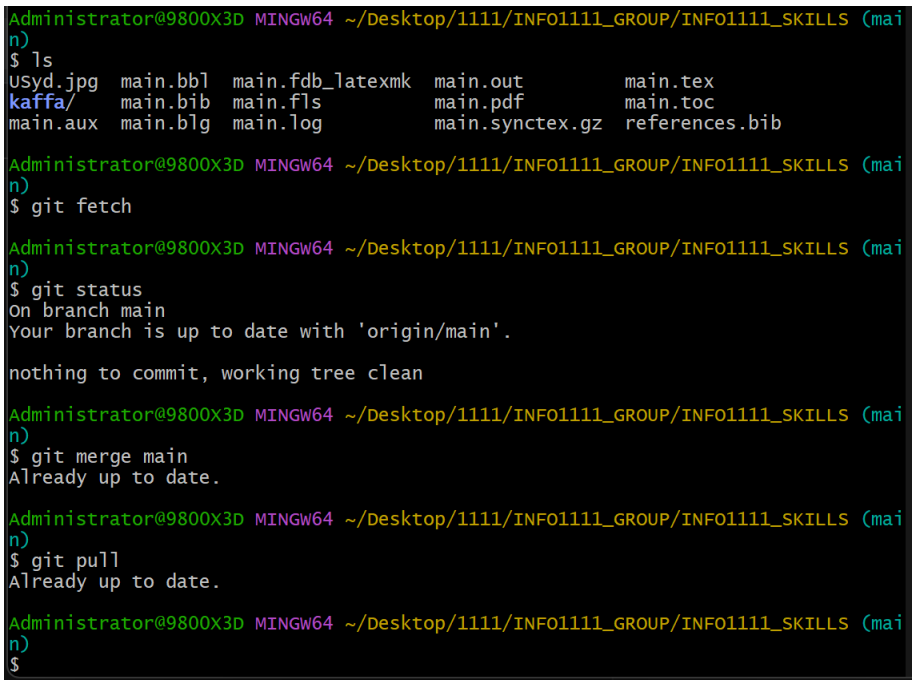


Figure 7: Pulling latest changes from remote repository

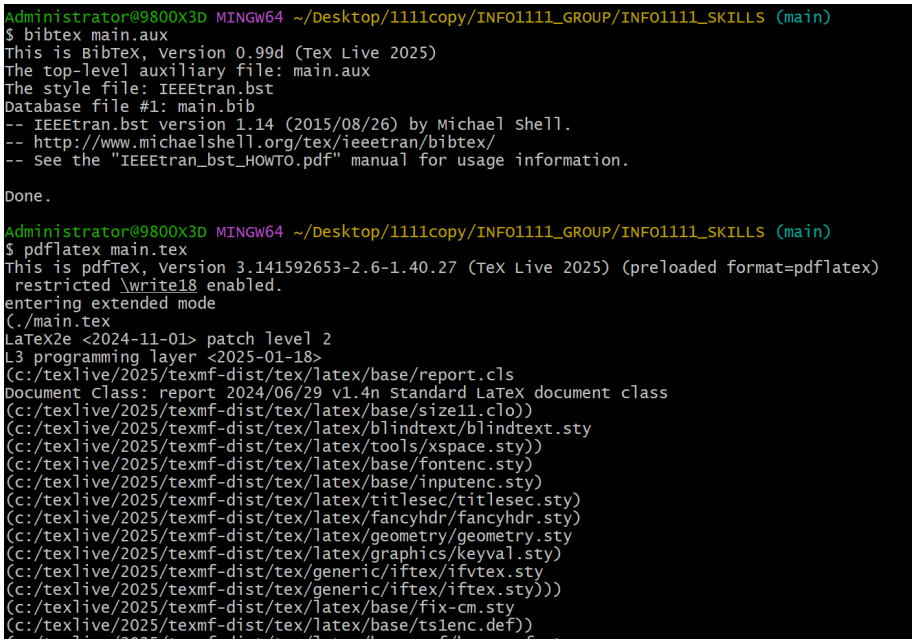
4. Generate PDF using Git Terminal

Compile your LaTeX document into PDF using Git terminal commands:

`bibtex main.aux and pdflatex yourfile.tex`

This generates a PDF document from your LaTeX file directly via terminal. Bibtex make sure citation correct

This picture illustrates generating a PDF using terminal:



2.3. Skills for SW Development: Jared Song

Through this project, I identified two critical skills from the SFIA framework relevant to software development:

Key Technical Skills

- **PROG (Programming/Software Development)**
  According to the SFIA framework [1], developing the disaster system's offline functionality required:
  - Implementing local data caching using Python's `shelve` module
  - Writing thread-safe code for concurrent access during emergencies
- **TEST (Software Testing)**
  Establish comprehensive test coverage for disaster scenarios123:
  - Parameterized test suite covering distinct failure modes:
    - Network partitions (simulated with `pytest-timeout`)
    - Data corruption (CRC32 validation tests)
    - Resource exhaustion (memory/stress tests)
  - Mock service framework featuring:
    - Configurable failure injection
    - Latency simulation
    - Stateful behavior modeling

Skill Development through Collaboration

According to the SFIA framework [1], The team environment enhanced these skills by:

- **Cross-domain feedback**: Data Science members' statistical analysis helped refine our cache invalidation algorithm, the data collected has also simplified the work and made the work more straightforward.
- **Collective problem-solving**: Pair programming sessions fixed race conditions in the resource allocator module
- **Tool knowledge sharing**: Learned GitHub Actions CI configuration from Computer Science teammate, which really helps me enhance my skills and understanding of GitHub.

Areas for Improvement

Through my work on the disaster response system, I've identified several technical and professional skills that require refinement:

- **Performance Optimization**: Need deeper understanding of profiling tools (e.g. cProfile) - evidenced when our stress tests failed at 10,000+ concurrent users, be able to learn more about Python and be proficient in using Python.
- **Technical Documentation**: Find difficulty in using Github and Latex, so learning more skills and implementing automated documentation generation are important.

Git Response

1. Clone the repository

Clone the remote repository to local computer:

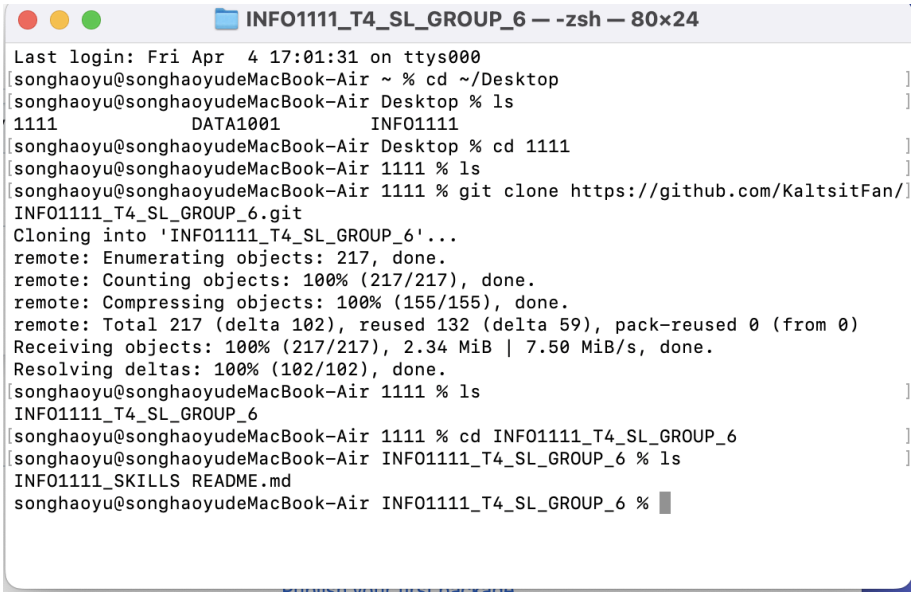
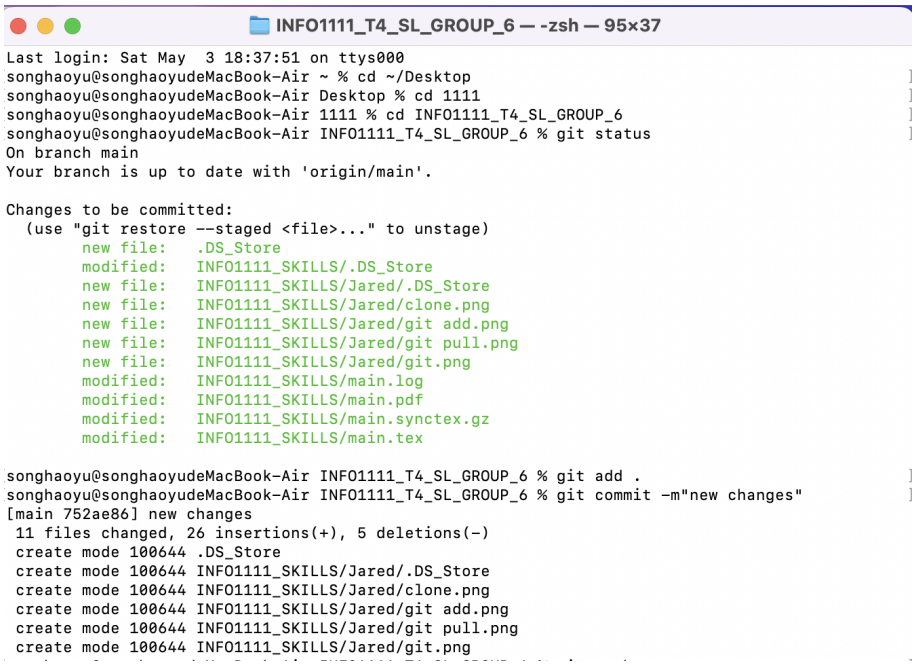


Figure 8: Cloning the repository

2. Commit and push changes to local repository




```
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git push origin main
Everything up-to-date
```

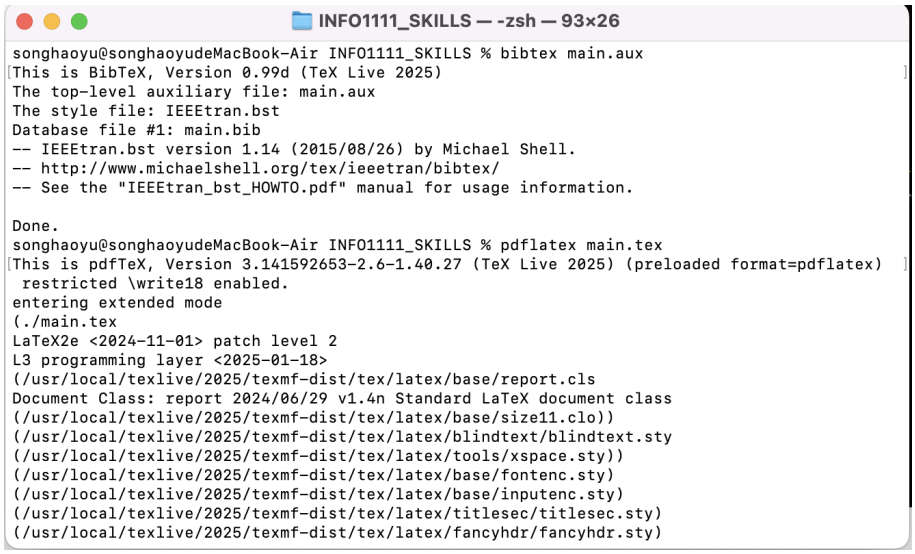
3. Synchronize repository

```
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git fetch
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 %  git merge main
Already up to date.
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 % git pull
Already up to date.
songhaoyu@songhaoyudeMacBook-Air INFO1111_T4_SL_GROUP_6 %
```

Figure 9: Pulling latest changes from remote repository

4. PDF Generation Through Git CLI



2.4. Skills for Cybersecurity: Barnett Zheng

Through this project, I identified two critical skills from the SFIA framework relevant to cybersecurity:

Key Technical Skills

- **SCTY (Network Security)** [4]
  Securing communication channels in the disaster response system required:
  - Implementing firewall rules to restrict unauthorized access
  - Encrypting data transmissions using TLS to ensure confidentiality
  - Deploying intrusion detection systems (IDS) to monitor for suspicious activity
- **SCTY (Data Protection)** [5]
  Ensuring the privacy and integrity of user data involved:
  - Utilizing AES encryption for sensitive personal information storage
  - Implementing access controls to restrict unauthorized data retrieval
  - Performing regular security audits to detect vulnerabilities

Skill Development through Collaboration

The team environment enhanced these cybersecurity skills by:

- **Interdisciplinary Insights**: Working with software engineers helped me align security mechanisms with application logic, ensuring seamless integration.
- **Incident Response Drills**: Collaborating with the team in security simulations improved my ability to detect and mitigate threats quickly.
- **Knowledge Sharing**: Gained practical experience with GitHub security features, such as dependency scanning and secret detection.

Areas for Improvement

Through my work on the disaster response system, I've identified several cybersecurity skills that require further refinement:

- **USUP (Incident Response)** [6]: Need to enhance my ability to analyze and react to security breaches in real time, particularly in high-pressure situations.
- **RESL (Risk Assessment)** [7]: Improve my ability to evaluate and prioritize security threats, ensuring that mitigation efforts focus on the most critical risks.

2.3.1. Git Response

1. Clone the repository

Clone the remote repository to your local computer:

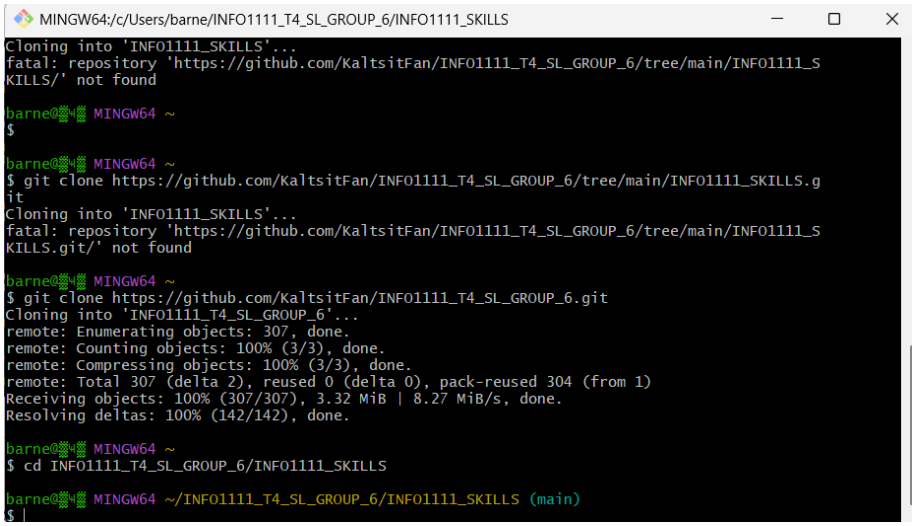


Figure 10: Cloning the remote repository

2. Standard Development Cycle

The regular workflow for making changes consists of these steps:

```
git add .                          # Stage all modified files
git commit -m "Descriptive message" # Commit changes locally
git status                         # Verify repository state
```

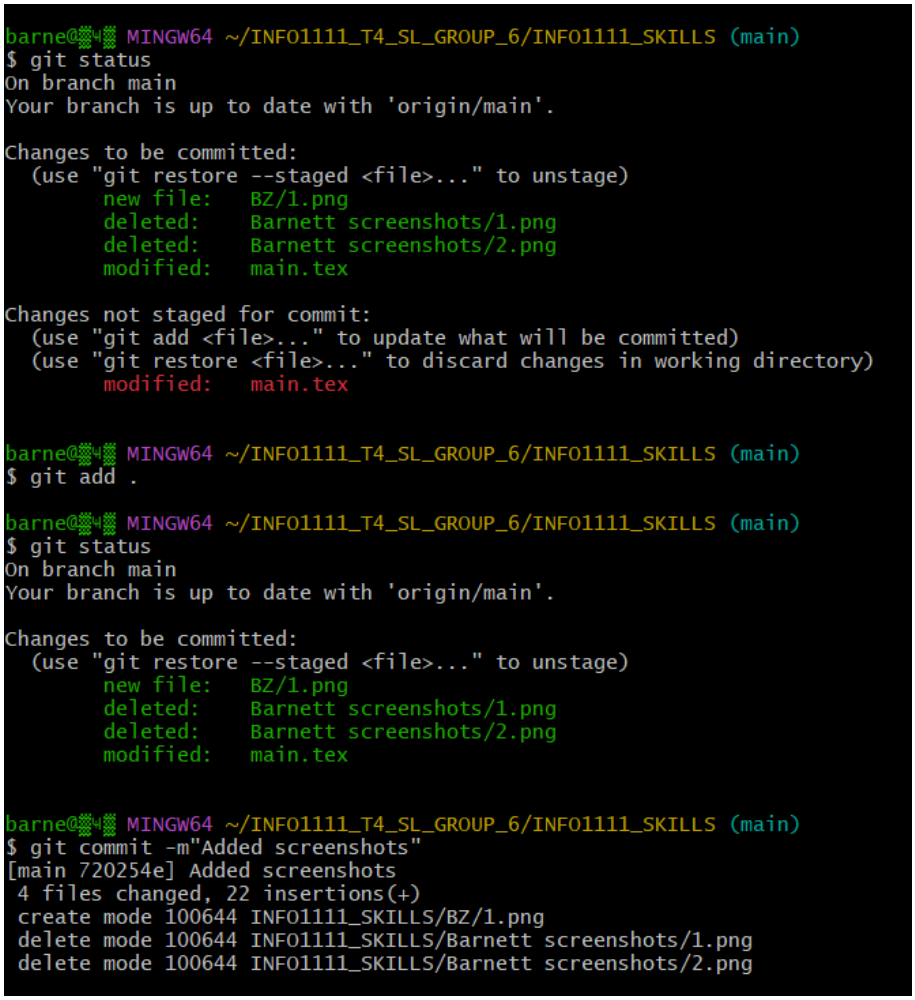


Figure 11: Git workflow: staging, committing

3. Document Compilation

Generate the project PDF through terminal commands:

```
pushing oringin main
bibtex main.aux      # Process citations
pdflatex main.tex    # Final compilation
```

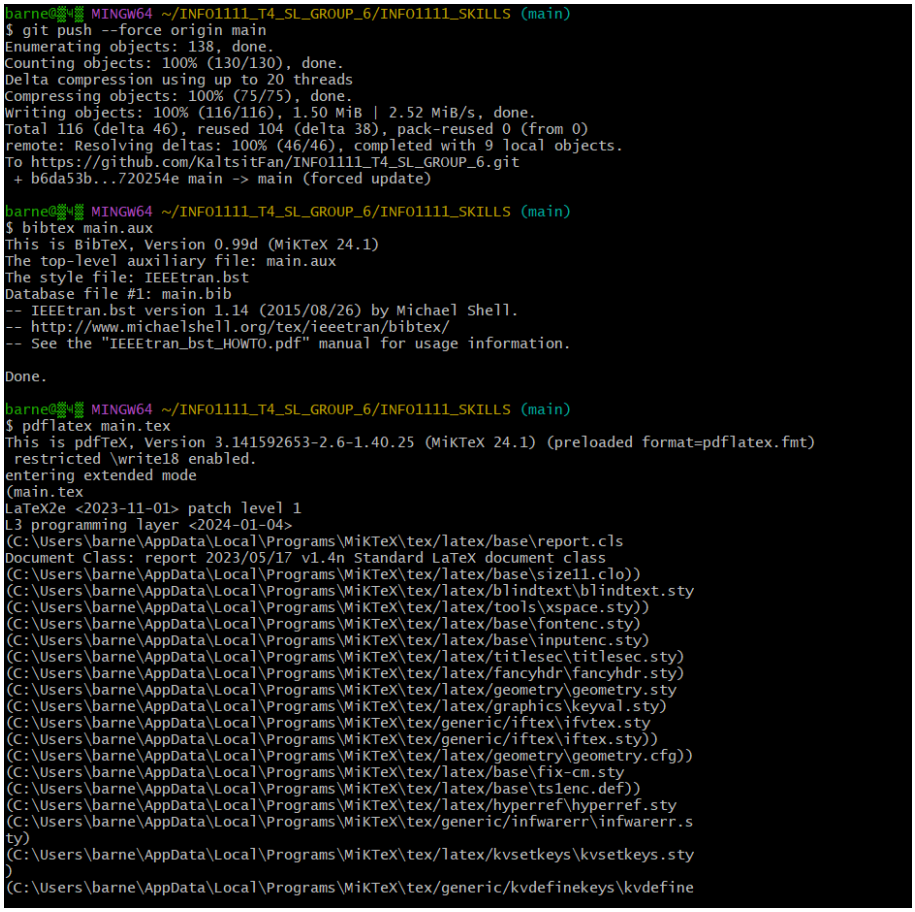


Figure 12: Compiling LaTeX document to PDF and pushing

Skills for Data Science: Link Lin

Through my work exploring data science applications, particularly in disaster response and predictive analytics, I identified two critical skills relevant to data science based on their practical importance:

Key Technical Skills

- **Data Integration**
  Effective data integration for disaster scenarios required:
  - Combining diverse datasets (e.g., satellite imagery, weather forecasts) using Python's `Pandas` library
  - Normalizing inconsistent formats for real-time visualization during emergencies
- **Predictive Modeling**
  Building robust predictive models for fire prevention involved:
  - Analyzing historical weather and topographic data with:
    * Machine learning algorithms (e.g., Random Forests)
    * Cross-validation to prevent overfitting
  - Simulating fire spread scenarios featuring:

* Temperature and humidity forecasting
* Wind speed impact modeling
* Risk area prioritization

2.3.2. git response

1. Clone the repository

Clone the remote repository to your local computer:

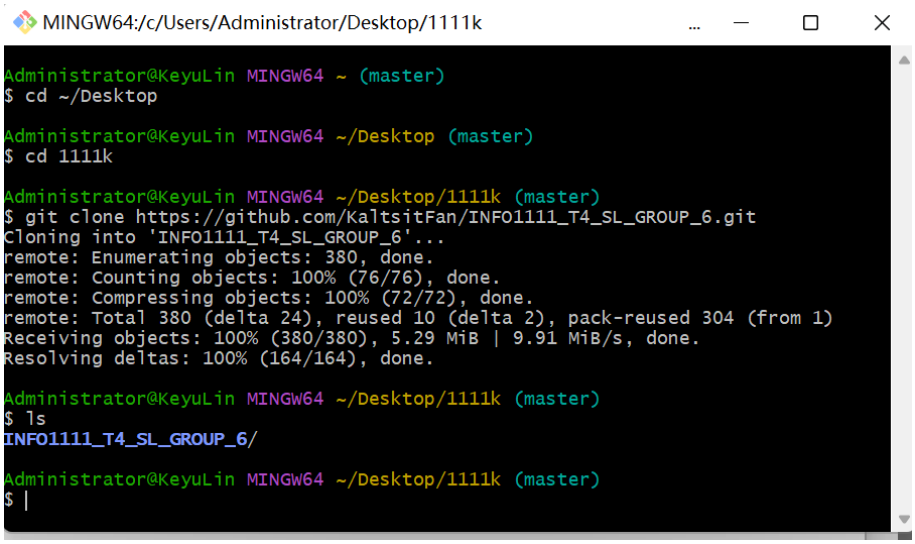


Figure 13: Cloning

2. Standard Development Cycle

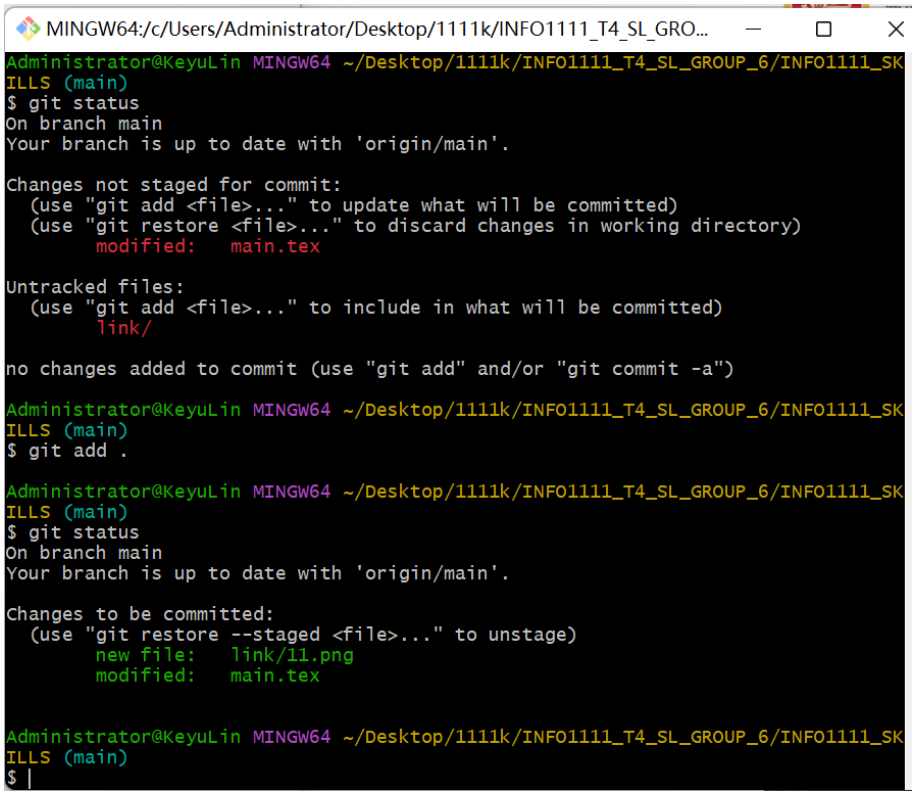


Figure 14: add

3. Testing and Deployment

Execute tests and deploy through terminal commands:

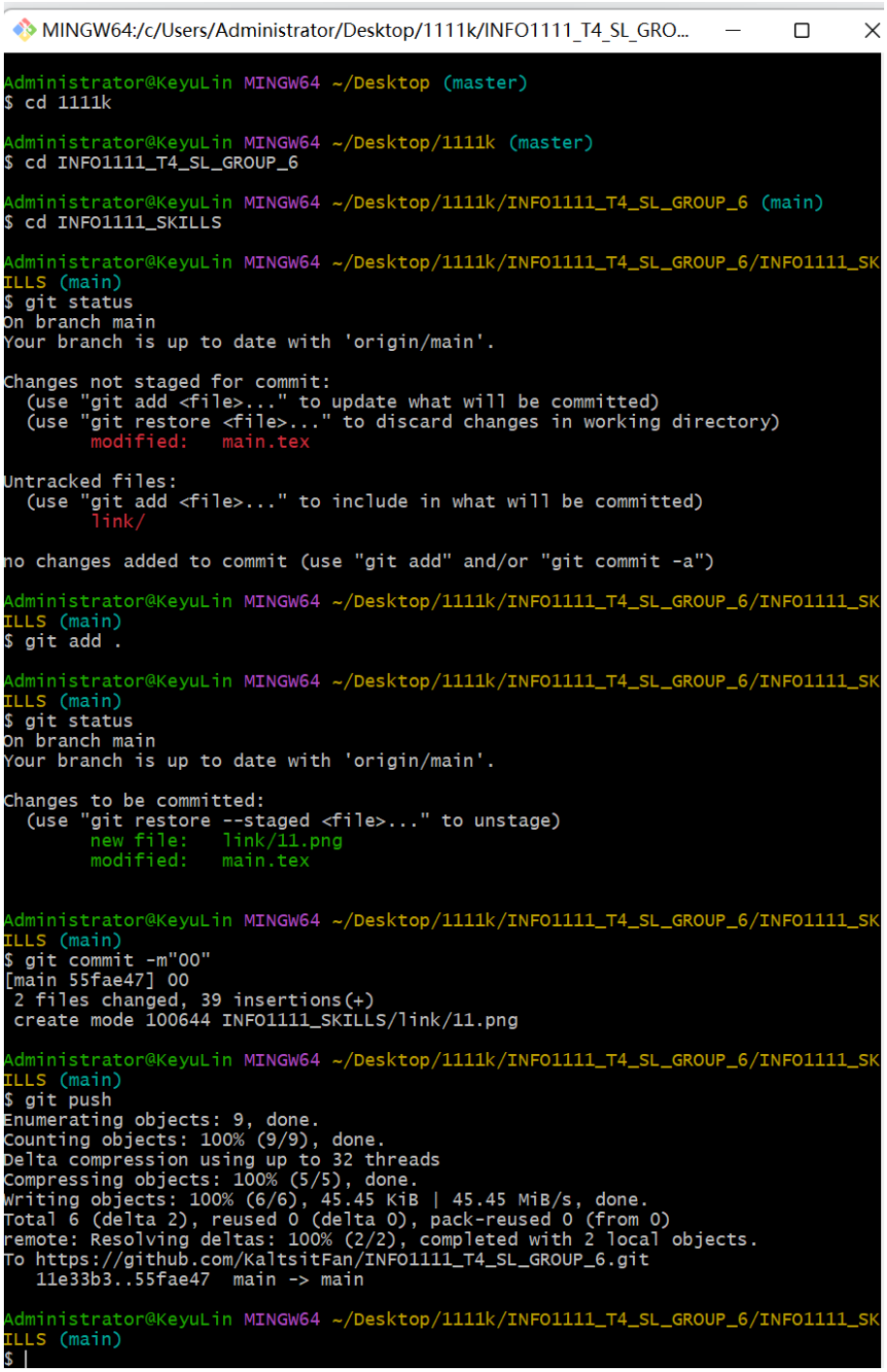


Figure 15: push

Skill Development through Collaboration

The team environment enhanced these skills by:

- **Cross-disciplinary input**: Feedback from software development teammates improved data pipeline efficiency, optimizing how integrated data fed into predictive

models.

- **Group troubleshooting**: Collaborative debugging sessions refined model accuracy by addressing data preprocessing errors.
- **Tool adoption**: Learned Jupyter Notebook workflows from a teammate, enhancing my ability to prototype and visualize data integration outputs.

Areas for Improvement

Through my data science efforts, I've identified key areas for growth:

- **Data Quality Handling**: Need better proficiency in manual data cleaning techniques (e.g., handling missing values), as shown when inconsistent weather data skewed early predictions.
- **Model Interpretability**: Struggle to explain complex model outputs clearly; improving visualization skills with tools like Matplotlib or Seaborn will aid communication with non-technical stakeholders.

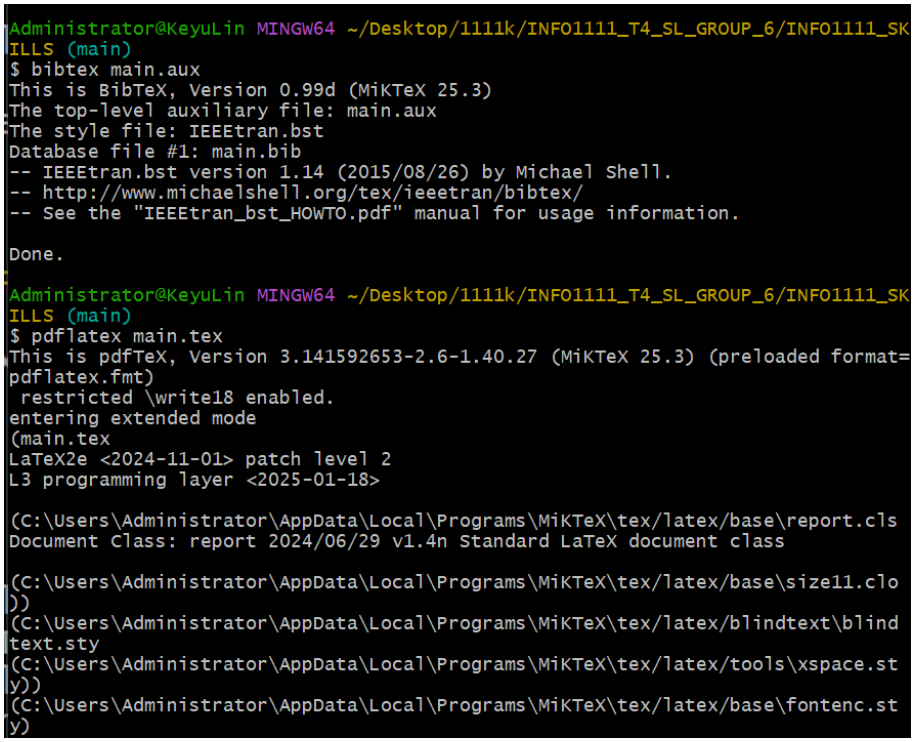


Figure 16: pdflatex

3. Submission contribution overview

For each submission, outline the approach taken to your teamwork, how you combined the various contributions, and whether there were any significant variations in the levels of involvement. (Target = ∼100-300 words).

3.1. Submission 1 contribution overview

As above, for submission 1 Kaffa DEmo2

3.2. Submission 2 contribution overview

As above, for submission 2

3.3. Submission 3 contribution overview

As above, for submission 3

Bibliography

[1] SFIA, "The global skills and competency framework for the digital world," 2022, see https://sfia-online.org/en/sfia-8/all-skills-a-z.

[2] Websockets Library Documentation, "Websockets library documentation, version 10.3," 2025, available: https://websockets.readthedocs.io/ (accessed May 21, 2025).

[3] SFIA Foundation, "Sfia 8 reference guide," 2021, available: https://sfia-online.org/en/sfia-8 (accessed May 21, 2025).

[4] National Institute of Standards and Technology (NIST), "Computer system security and privacy advisory meeting," 2002, see https://csrc.nist.gov/CSRC/media/Events/CSSPAB-JUNE-2002-MEETING/documents/Lainhart-06-2002.pdf.

[5] European Union Agency for Cybersecurity (ENISA), "A trusted and cyber secure europe," 2024, see https://www.enisa.europa.eu/sites/default/files/2025-02/A%20Trusted%20and%20Cyber%20Secure%20Europe%20-%20ENISA%20Strategy%20-%20Indicators.pdf.

[6] SANS Institute, "Incident response," 2021, see https://www.sans.org/security-resources/glossary-of-terms/incident-response/.

[7] International Organization for Standardization (ISO), "Information security, cybersecurity and privacy protection," 2022, see https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27005:ed-4:v1:en.