Platform Information – 3/24/2025 Updated

1. Overview

- Project Name: Chess Plug
- Description: Chess Plug is a team based game, designed specifically for MSUM students and on the smoke Linux machine. Teams are semi-randomly generated, using players analytics and initial hard coded players to split and set up teams. All Data for game is stored on a GitHub that only a specific account has access to change or edit. A website, utilizing Microsoft OAuth2 login will allow students of MSUM not in smoke to better see the chess board and make votes.
- Contacts:
    - Judah Nava: 971-283-9880, judah.nava@go.mnstate.edu
    - Laween Al Sulaivany: laween.al-sulaivany@go.mnstate.edu
    - Ben Johnson: benjamin.johnson@go.mnstate.edu
    - James Zook: james.zook@go.mnstate.edu
    - Donovan Morse: Donovan.morse@go.mnstate.edu
    - Ugnius Bieliunas: ugnius.bieliunas@go.mnstate.edu
    - Jonathan Moritz: jonathan.moritz@go.northlandcollege.edu

2. Application Stack

- P. Languages and Frameworks:
    - Python (3.10.12 – smoke current install)
    - Flask – for hosting the chess website
- Pip Libraries/Dependencies:
    - chess
    - PyGithub
- Databases:
    - Github
- API's and third-part integrations
    - Microsoft auth-oauth2
- Build tools and package managers
    - Pip

3. Target Environment

- Operating System: Linux, Ubuntu 22.04
- Runtime Environment:

    - Python 3.10
    - Web server – Apache (smoke infrastructure)
- Hardware requirements: NA
- Deployment model:
    - On-premises: All backend services and website will be deployed to smoke.
    - Code Repository: All source code, documentation, and static data will be stored on GitHub (same repo as data).
    - Potential External Hosting: In the event of hosting the website is not supportable on smoke, the website can be hosted on chess.judahsbase.com

4. Infrastructure

- Environments (Dev/Test/Prod):
    - All development, testing, and production use the same environment.
    - Testing is integrated into the production code and can be toggled using a command-line argument or runtime flag.
    - No separate stating or QA servers are used.
- CICD Pipelines:
    - No automated CI/CD pipelines are in place.
    - Code is manually built, tested, and deployed as needed.
    - Future considerations: explore lightweight CI (e.g. GitHub Actions) if automation becomes beneficial.
- Source Code Hosting:
    - Source code is hosted on GitHub in private repository.
    - Git is used for verions control, with branching and tagging used for managing releases.
- Monitoring and Logging Tools:
    - All monitoring and logging functionality is custom-built as part of the project codebase.
    - No external monitoring or log aggregation services are used.

5. Security & Access

- Authentication method:
    - Website: OAuth will be used to authenticate users (Microsoft)
    - Smoke: Users are authenticated through existing Linux user accounts.

6. Questions for IT

- Can a central account for chess be set up in smoke?
    - Host the website, own the cron, store the chess command?
- Does IT need anything regarding Monitoring and Logging?
- What are the expectations for addressing vulnerabilities?
- Is there an alternative solution if utilizing venv is not allowed?
- Is there specific documentation that IT would need about it?
- Does smoke have a schedule if and when it shuts down / restarts?
-