List of Detailed Contributions by Each Team Member

Orion Gregory, Sebastian Shirk, Tejas Bhadoria

Orion:

- **Multithreaded/asynchronous tournaments:**
  - Implemented Celery into the tournament code to run bots asynchronously for a fast runtime.
- **Leaderboard system/Scores:**
  - View logs and download PGNs
- **Teachers can revisit old tournament logs:**
  - Teachers can revisit old tournaments and get their logs
- **Docker and Hosting:**
  - Get working docker container
    - Working with Celery, Redis, and Django. Configured PSQL connection alongside Django and Docker
  - Host our application
    - Utilized Google Cloud Run to host application
      - Used previously build docker container
      - Testing with different CPU, Memory configurations for optimal performance with Bot Multithreading
- **Wrote bots for testing:**
  - Wrote a multitude of chess bots that were used for testing web app features.
    - Includes bots written with and without Stockfish
- **Tournaments:**
  - Updated and refined tournament functionality
    - Upload multiple bots and play them all against each other in a round robin style.
    - Updated tournament logic to allow all types of bots, not just Stockfish
    - Got tournament logs up and running, providing feedback as to how the game played out.
- **Dockerfile update:**
  - Updated docker files to work for our updated backend
- **Documentation**
  - Writing installation, development, usage documentation
  - Wrote Github pages documentation

- **Sprint Deliverables:**
    - Sprint 1 Deliverables

Sebastian:

- **Initial backend:**
    - Dropped the first backend push.
        - Upload bots
        - Play against bot
        - Communicate with basic frontend
- **NavBar skeleton:**
    - Basic frontend update to add a NavBar.
- **Local PostgreSQL:**
    - Local PostgreSQL implementation to test registration/login and file uploading before cloud implementation.
        - Creating local databases
        - Running queries to gather data from our dbs
- **Teacher CRUD:**
    - Teachers have the ability to edit students and student bots.
        - Delete students
        - Delete student bots
        - Edit student bot variables
            - Name
            - Description
            - Visibility
            - Status
- **Update student features:**
    - Assigned: Sebastian
    - Limit student features to just upload, activate, archive.
- **Teacher tournament dashboard:**
    - Took the existing teacher tournament dashboard and made it cleaner.
        - Moved tournament creation to a pop-up rather than being a constant thing on the page for a cleaner and more organized look.
    - Fixed tournament deletion
    - Fixed tournament status not updating properly
- **Remove redundant features/adding frontend accessibility options:**
    - Remove unnecessary or redundant features.

        - Removed “classes” feature
        - Removed “password/account management”
        - Removed excessive ways to get to the tournament creation
        - Removed “cancel tournament” feature. It was redundant due to “delete tournament”
        - Removed student ability to see other student bots that aren’t public
    - Add more quality of life features
        - Added a way to bulk select/upload bots to a tournament.
        - Clicking “Dashboard” from tournament view takes you back to tournament section of teacher dashboard rather than defaulting to student section of teacher dashboard.
- **Leaderboard system/Scores:**
    - Show scores of participants after a tournament
        - Added “Recalculate Scores” button.
    - Implement leaderboard section of the teacher dashboard
        - Leaderboard shows compilation of every bot that has participated in at least one tournament.
        - Shows statistics including games played, games won,games drawn, games lost, win percentage, draw percentage, number of tournaments participated in, leaderboard rank, bot name, and bot owner
        - Added dropdown that gives the option to see stats for all tournaments or just specific tournaments
- **Sprint Deliverables:**
    - Sprint 2 and 3 deliverables
    - Project Video
- **Documentation:**
    - Edits to installation, development, usage documentation

Tejas:

- **Update React frontend to reflect backend changes:**
    - Update current frontend to reflect changes made in backend
    - During refactoring of project, we focused towards using html with bootstrap instead for backend production

        - Moving from bootstrap HTML to react frontend
    - Created initial React frontend for basis of project
- **OAuth implementation:**
    - Created OAuth client
        - Configured callback
        - Created environment variables for client secret and key
        - Created OAuth login page
- **Original Bot vs Bot Implementation:**
    - Created Bot vs Bot code that acted as foundation of Tournament code
- **Wrote bots for testing:**
    - Experimenting with chess bots that are functional without Stockfish
- **Setting up google cloud environment:**
    - Created PostgreSQL server and configured it for development use
    - Created OAuth Client within Google Cloud and configured it for development use
- **Docker:**
    - Configured Docker containers to work with React frontend server alongside backend python server
- **Documentation:**
    - Worked on installation, development, usage documentation