# Web-based Chess

Patrick Lawrence, Trush Patel, Ishmail Koroma, Kevin Codd, Matt DiStefano

# Table of Contents

# Author Resumes

# Kevin Codd

## Data Scientist // Software Engineer

kncodd@gmail.com
+1 (484) 947- 9711
github.com/KevinCodd
linkedin.com/in/kevin-codd
Greater Philadelphia

## EXPERIENCE

**Data Science Intern** — Jun 2021 – Present
CellCarta — Remote (Fremont, CA)

- Researched and improved upon multivariate time series outlier detection algorithm for Flow Cytometry data
- Developed data pipelines in Python and R for comparative evaluation of outlier detection algorithms on many datasets
- Contributed to R API toolkit for CellEngine data analysis software

**Data Science Intern** — Feb 2020 – May 2021
nth Solutions — Exton, PA

- Developed Java backend for IMU sensor data visualization application
- Modelled time series data using several Python and MATLAB libraries for state space modelling, deep learning, and frequency domain analysis
- Researched and implemented IMU sensor fusion algorithm for accurate orientation tracking
- Led team of interns responsible for data engineering and analysis tasks

**Undergraduate Research Assistant** — Spring, Fall 2021
West Chester University — West Chester, PA

- Analyzed break-in attempts to Computer Science department's servers using Linux authentication logs and PySpark in support of academic research paper
- Cleaned, parsed, and analyzed logs using PySpark to identify and quantify attacks
- Currently developing backend for data-streaming dashboard using Spark and Kafka

**Regulatory Affairs Intern** — May 2018 – Nov 2018
Lungpacer Medical — Exton, PA

- Organized clinical trial data and produced reports analyzing statistical trends
- Conducted scientific literature searches to substantiate efficacy of medical device treatment approach
- Proofread and edited technical documentation to be submitted to regulators

## EDUCATION

**B.S. in Computer Science & Minor in Mathematics** — Expected Dec 2022
West Chester University

- **GPA**: 3.95 / 4.0
- **Relevant Courses:** Big Data Engineering, Mathematical Statistics, Artificial Intelligence, Software Engineering, Data Structures & Algorithms, Computer Systems, Linear Algebra, Multivariable Calculus

## SKILLS

- **Languages:** Python, Java, R, C, MATLAB, SQL, SAS
- **Tools/Frameworks:** *Data analysis/engineering*: NumPy, SciPy, Pandas, Matplotlib, PySpark, Kafka. *Machine learning*: Scikit-learn, TensorFlow. *Development*: Git, Jupyter Notebook, Anaconda.

## ACTIVITIES

**Competitive Programming Club** — Aug 2020 – Present
West Chester University

- Club President (Aug 2021-Present) - Facilitated club meetings and competitions, created and curated practice material
- Algorithmic programming contests: member of team advancing to 2021 ICPC North American Division Championships, 3rd place team out of 24 in CCSC Eastern 2020 Programming Competition

# Matthew C DiStefano

13mdistefano@gmail.com

*805 Windridge Lane | Downingtown | PA | 19335 | tel: 610.247.8772*

## Education

**West Chester University of Pennsylvania**

Bachelor of Science in Computer Science

3.64 GPA | Expected to graduate May 2022

**Relevant Courses**

Computer Systems, Data Structures and Algorithms, Calculus, Database Management Systems, Data Communication and Networking, Edge Computing and Deep Learning, Software Testing, Intro to Cloud Computing, Digital Image Processing

## Professional Experience

**Law Clerk | June 2014 – August 2017**

**Dalton & Associates, LLC | Wilmington, DE**

– Managed and maintained the systems for closing of files and server storage

– Maintained and kept track of client's medical bills and records in accordance with HIPPA standards for ease of access by attorneys and clients alike

– In charge of maintaining files within our cloud system

**Law Clerk | June 2019 - Present**

**Potamkin ARM, LLC | Downingtown, PA**

– Keep over 170 entities up to date and manage documents that are to be kept in the minute books

– Maintained timely communication with fellow employees to better identify and resolve issues

## Computer Science Projects

**CSC496 Final Project | West Chester University | December 2021**

– Lead a team to the creation of a graph based shortest path algorithm used to determine the earliest semester a specified Computer Science course could be taken at WCU

## Skills

– Languages: Java, JavaScript, C#, Python, SQL, HTML, OCaml

– Tools/Frameworks: Git, Node.js, React.js

# Patrick Lawrence

West Chester, PA
(610)-675-8651
patrickjameslawrence@icloud.com

## Education

West Chester University of Pennsylvania
*Bachelor of Science in Computer Science* | May 2022

- 3.38 Cumulative GPA

## Related Experience

*IT Helpdesk Consultant* | December 2020 – Present
West Chester University of Pennsylvania | West Chester, PA

- Identify user issues and dispatch them to appropriate teams when needed
- Educate fellow consultants on internal IS&T functions and solutions to issues users face
- Maintain the IS&T knowledge base; adding articles to pass down my expertise when needed
- Maintain cordial communications and de-escalate situations with bad-tempered users

*Software Engineer Intern* | June 2021 – July 2021
Universal Health Services (UHS) | King of Prussia, PA

- Built new FTP connections to transfer patient records, test results, and other data between UHS network hospitals
- Utilized SQL and C# to create applications in .NET
- Maintained adherence to all HIPPA privacy standards and protected patient data
- Shadowed UHS software engineers in their meetings and projects
- Reviewed standard UHS operating procedures, all internal IS&T teams and their functions, and the software development lifecycle
- Learned office etiquette and best practices from other employees in the office

## Skills Summary

- *Programming Languages:* Java, C++, JavaScript, C#, SQL
- *Development Tools:* Version control & bug tracking
- Heavy interest in advanced mathematics

## Projects

*Synth* | September 2020 – March 2021
Multi-function Discord bot | Node.js and MySQL

- Created a robust command and event handler
- Designed a local MySQL database to store various configuration information from servers and users

*Keychain* | October 2020
Social media sharing service | Android Studio and MongoDB

- Designed an Android application using Android Studio
- Utilized database services to create a secure account login solution

# Trush D. Patel

4025 Hunt Drive, Doylestown, PA 18902 | (267) 261-9675 | trushp3@gmail.com| https://github.com/trushpatel

## EDUCATION

Aug 2020 - Present **West Chester University of Pennsylvania - West Chester, PA**
Bachelor of Science Dual Major in Computer Science and Finance
GPA 3.99, Expected to graduate in May 2022

## WORK EXPERIENCE

Dec 2021 - Present **Tutor - West Chester University of Pennsylvania**

- Organized one-on-one meetings to assess comprehension and teach fundamental concepts
- Instructed the development of effective time management and test preparation strategies
- Tutored undergraduate students in Introduction to Statistics, Foundations of Computer Science, and Data Structures and Algorithms
- Acquired CRLA Level I certification for professional tutors and peer educators

Oct 2021 - Present **Research Assistant - Dr. Jongwook Kim**

- Researching software design patterns and refactoring paradigms
- Assessing program transformations that augment software reliability and extensibility
- Building tools to invoke Eclipse refactoring through the command line
- Automating tests for efficacy of refactoring tools in different IDEs

## PROJECTS

Oct 2021 **Yelp Dataset - Python|Apache Spark**

- Built an algorithm by implementing PageRank to identify influencers in a 10-gigabyte dataset
- Computes the top 5% of influential users from quantity and quality of reviews and connections
- Extracted each user's review data to identify individual preference in type of places reviewed
- Pinpointed reviewed business locations to cluster users geographically

Sept 2021 **Drowsy Driver - Python|Arduino Uno**

- Created a driver engagement program to detect whether a driver is drowsy
- Implemented facial landmarking to identify key indicators of drowsiness
- Modeled a linear SVM classifier that predicts user's drowsiness with 95% accuracy
- Delivers haptic feedback to alert driver through vibration motors

Aug 2021 **A* Pathfinder Algorithm Visualizer - Python**

- Created a program that visualizes the A* pathfinding algorithm
- Procures end points and obstacles to visualize the optimal path in a grid
- Calculates the optimal path with a combination of uniform cost and greedy search

## EXTRA-CURRICULAR ACTIVITIES

Sept 2020 - Current **Competitive Programming Club - Programmer**

- Developed efficient algorithms to solve programming problems
- Mentored other team members on problem-solving techniques and program design
- Won 3rd place in the PACISE Regional Programming Competition

Sept 2020 - May 2021 **Investment Group - Fund Manager**

- Managed an investment portfolio with an aggregate value of $60,000 achieving 22% growth
- Created pitches for stocks to be purchased for growth and value funds
- Designed and led virtual workshops teaching financial literacy and equity valuation to implement analytics-oriented strategies

## HONORS AND AWARDS

December 2021 **Upsilon Pi Epsilon - International Honor Society for the Computing and Information Disciplines**

April 2021 **3rd Place - PACISE Regional Programming Competition**

Dec 2020, May 2021, Dec 2021 **Dean's List - West Chester University of Pennsylvania**

## SKILLS

| | |
|---|---|
| Languages | Java, Python, C++, C, JavaScript, HTML, CSS, OCaml, Bash, SQL |
| Tools/Frameworks | Ansible, Apache Spark, Bootstrap, Docker, Git, Jenkins, Linux, Kubernetes, PyTorch, React.js |
| Coursework | Artificial Intelligence, Big Data Engineering, Computer Systems, Data Structures and Algorithms, Introduction to Cloud Computing, Linear Algebra, User Interfaces, Software Engineering, Software Testing |

# Morlai Ishmail Koroma

106 Norma Road, Yeadon PA 19050 ishmailkoroma1@gmail.com (267) 515 2702

## EDUCATION

**West Chester University** Expected Graduation Date: May 2022
700 S High St, West Chester, PA 19383
Major: Computer Science

## TECHNICAL SKILLS:

- Proficient in JavaScript, C++, Python, HTML, Linux, and Microsoft Office

## PROFESSIONAL EXPERIENCE:

**U.S. Navy** **Naval Research Enterprise Intern**
Philadelphia, PA October 2021 – December 2021

- Researched and provided recommendations and new methods for increased monitoring, reducing maintenance, and repair of critical naval machinery systems on Navy ships.
- Provided a possible application of LoRa sensors on the Lube Oil System on Navy ships to detect water leaks, maintain temperature pressure and reduce the maintenance cycle for sailors.
- Developed a cost analysis framework for the materials used in the implementation of the LoRa sensors on the Lube Oil System.

**U.S. Navy** **Naval Research Enterprise Intern**
Philadelphia, PA June 2020 – August 2020

- Researched methods to enforce LoRa (long-range) technologies on Naval platforms without compromising security.
- Assessed the LoRaWAN (long-range, wide area network) vulnerabilities and cyber security risks involved in implementing long range sensors on Naval platforms

**Avista Healthcare** **Dietary Aide**
Cherry Hill, NJ August 2019 – March 2020

- Monitored inventory and stock for kitchen ingredients
- Maintained kitchen equipment and appliances via guidelines of the facility's dietary manager and kitchen cook
- Oversaw take-down of dining areas, collecting all silverware, discarding of leftovers and removal of garbage and recycling items.

**Sky is the Limit (Moving Company)** **Moving Assistant**
Greater Philadelphia June 2018 – May 2019

- Tracked payments for services
- Kept detailed inventory of customer items to ensure safekeeping and integrity of items moved.
- Kept track of communication and interaction with all customers via emails, texts, and calls

### COMPUTER SCIENCE PROJECTS:

**CSC 472 Final Lab Project** **Lead Member**
West Chester University November 2021 – December 2021

- Performed a successful multi-stage exploit attack on a file to get and reveal content inside the target flag.
- Completed format string, ROP, GOT overwrite, and stack overflow attacks on the program to leak specific information.

### CLUBS & VOLUNTEER EXPERIENCE

**WCU Weekly** **Videographer/Editor**
West Chester University August 2020- December 2020

- Filmed highlights at West Chester University sports games and helped edit weekly sport segments

**Christ The King Prayer Chapel.** May 2018 to August 2018
Philadelphia, PA

- Filmed services.
- Helped organized the church and collected donations/offerings from members

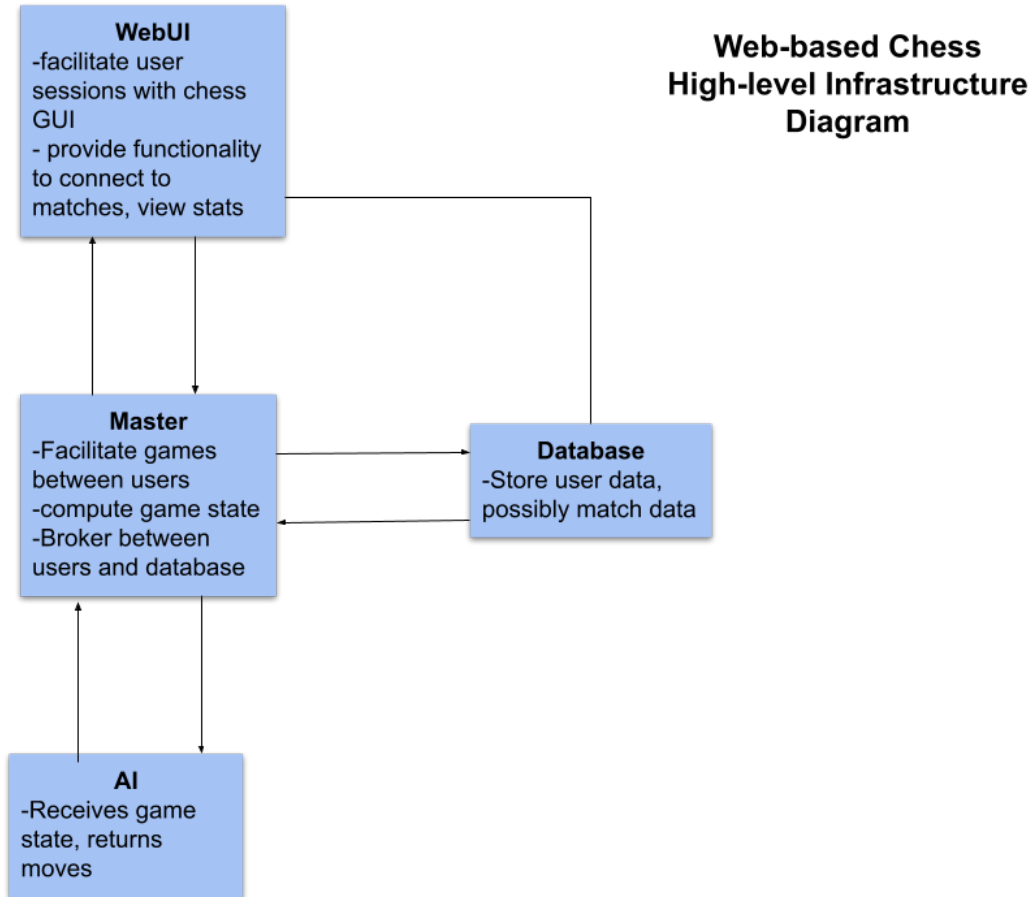
### INTERESTS:

- Cybersecurity; Network and Computer Systems; Artificial Intelligence; Computer Graphics; Video Games; Basketball; Football

# Technical Report

## Chapter 1 -Project Vision

Our overall vision for this project is to create a cloud-based chess game playable in the browser — essentially like a simplified version of chess.com or lichess.org. Core functionality would include a fully-implemented GUI-based chess game with options to play against a human or AI opponent, creation of user accounts and persistent storage of user data, and matchmaking between users. Below is a high-level diagram of the rough infrastructure we envision.




## Chapter 2 - Technical Implementation

### Web UI

The web UI's key functionality will be GUI-based chess. The user will be able to play chess on a graphical chess board, most likely with a point-and-click control paradigm. The web UI will also include pages for login and account creation, and for viewing user stats. Some candidate frameworks for

implementation include JavaFX, React.js, and Vue.js, but we must consider which will provide the most efficient rendering and convenience for the user.

## Key Features

| **User Actions** | **Potential Implementation** |
|---|---|
| Create/update/delete account | When a user first connects to a webUI session, there will be a button linking to a sign-up window where the user can create a username and password. We can also have a "user settings" button somewhere in the primary window that brings up options to update username/password or delete an account. |
| Login | When a user first connects to the web UI, they will be presented with a window with options to enter their username and password or create an account via a button. |
| View stats | A button will link to a window where a user can pull up basic stats such as win/loss ratio, and their game history. This data would be read and calculated from the database. |
| Create invitations and join matches | Users can choose to create an invitation - either public or to a specific user. A list of invitations is then periodically published to all connected web UI sessions with permission to view them, and other users can accept these invitations. The user can also choose to create a game against an AI rather than play against other users. For all of these options, the user must also specify the starting amount of time for the match timers. (2, 5, 10 minutes) |
| Move pieces via point-and click/drag and drop | In the gameboard GUI, users can move pieces by dragging and dropping. A background rule-checking function will verify that the move is valid and allow it to be executed on the GUI if it is legal. |
| Game clocks | Clocks for both players will display how much time they have left. Time controls can be customized, perhaps via a drop-down when game invitations are first created. |
| Resign/leave a match | There will be a button to resign/leave a match, ending the game session in a win for the resigner's opponent. |

| Backend Functionality | Potential Implementation |
|---|---|
| Check that moves are legal before allowing execution. | This legality check could also run on the master node, but implementing it on the webUI node would likely reduce lag. In either case a function would take in the current game state and proposed move and ensure it is in compliance with all rules. |
| Send moves to “master”. | Once a legal move is executed, it needs to be sent to the master to update the opponent’s web UI. |
| Send and receive game invites. | When a user creates a game invite, it must be sent to the master node to be pushed to all other users with viewing permissions. Similarly, the WebUI must be open to read/receive a list of open invites. This could potentially be done through some kind of data streaming framework such as Kafka, or by writing to and reading from the database. |

## Master

The “master” comprises a few different functionalities, and will primarily serve as a dedicated server for matches between users.

The master will facilitate game sessions between users and store game state independently of web UI sessions. Users in web UI sessions will be able to submit game invitations (perhaps public or to a particular user) which the master will then display in a listing in other web UI sessions. Other users can accept the invitation, prompting the master to create a game session. When a user makes a move, the master stores the updated game state and sends it to the other user’s web UI session. It will also compute rating changes (if implemented) after a game and write updated game history and user stats to the database. Alternatively, game state could be written to and read from the database in real time while the game is in-progress.

To facilitate matches between more than a single pair of users, the master could keep a registry of users logged into webUI sessions, and keep track of which ones are already paired into a game session. The master would track the game state for each game. Most of this functionality could likely be implemented using Java.

### Key Features

| Functionality | High-level Implementation |
|---|---|
| Keep registry of users connected to web UI sessions | The master will keep track of all users that have connected to the network with their login as well as the players that are connected and currently playing a game. |

| | |
|---|---|
| Push list of game invitations to users | When a game invitation is created, the master will notify other players that there is a game ready to join. If the invite is public, all users will see the invitation. If the invite is to a specific user, only that user will see the invitation. |
| Track game state for all in-progress games | The master will store the sequence of moves made so far and which player's turn it is for each game in memory. |
| Timeout/auto-resign | Should a user have connection issues or disconnect from the application, there will be a small amount of time to allow the user to reconnect and resume play. When logging back in, the master will check if the user is supposed to be in a game and automatically connect them and play will resume. Should the user not connect in time, they will automatically resign. |
| Facilitate matches | The master will serve as a broker for game invitations, creating a "session" between two players or a player and AI when a game is initialized. It will track game state, receive moves from players and push those to the other player's web UI session, serve as the official timekeeper for the match, tell users who they are playing against, and end the session once it ends in a win, loss, draw, or disconnection. |
| Initialize a game | When a game between two players is created via an accepted invite or a request to play against the AI, colors will be randomly assigned and the clock will be started. |
| Timers | The master will keep the official match time and write it to the connected web UIs periodically. |
| Write results and updated player stats to database | At the end of the game, the game could be encoded in a file format such as FEN or PGN and written to the database. Players' game history could also be updated. |
| Update player ratings (tentative) | We could implement an elo rating system where at the end of every game the master calculates rating updates and writes them to the database. |

## Database

The database will store user account information including username, rating and game history. This could be implemented using several different database frameworks, including mySQL, Redis, and MongoDB. A relational database would likely serve our needs for this project. For example, this could be useful for searching game history by player.

## AI Chess Player

We'd like the user to have the option to play against an AI chess player should they want to. Implementing an AI is an important stepping stone for us in this project because it's very likely that there will not be any users to matchmake with at times. Having the AI player will allow these users to use the site during these slower times.

Behind the scenes, this AI will receive a game state and return what it believes to be the best possible move for each board position. A feature like this could be implemented in a couple different ways. We could bite the bullet and implement our own AI using the well-known minimax algorithm. This is a recursive or backtracking algorithm used in decision-making and game theory. It provides an optimal move for the player assuming that the opponent is also playing optimally. Alternatively, our other option is utilize open source options that already exist such as Stockfish to do this task for us.

Stockfish is one of the most well-known chess engines available to the public and it's considered one of the strongest as well. Many chess players consider the engine to be stronger than human players at the game. It would be able to provide better moves for board positions than our algorithms could and it would greatly increase the quality of the user experience.

In either case we would need to implement fine tuning the AI difficulty to allow users of all skill levels to play against the AI. This would likely involve giving it a probability of playing varying degrees of suboptimal moves.

We are currently leaning towards using Stockfish, as creating our own AI would be way too time-intensive for this project. We've decided to consider adding our own AI algorithms as a "stretch" goal for us to experiment with once other key functionality is complete.