**HACKATHON**

**TECH ADRISHTA**

**Project Report : Check & Mate**

**Dated**: 1st Dec 2020

**Team Number**: 29

**Team Members**:

1. Mayank Pagar
2. Annirudh VSP
3. Harsh Kumar Pandey

**Introduction**

We chose the problem Check & Mate due to our shared enthusiasm for chess as we have also started playing a lot of chess this lockdown. We also were thrilled to make an application that would help us get better at the game.

For this application we used the Flask Framework as it is a very lightweight and powerful tool for building web apps and we could also integrate the python backend, a language with which we are very comfortable with.

**Project Overview**

We have tried to create an application that helps chess players newbies and seasoned veterans alike to improve their game.

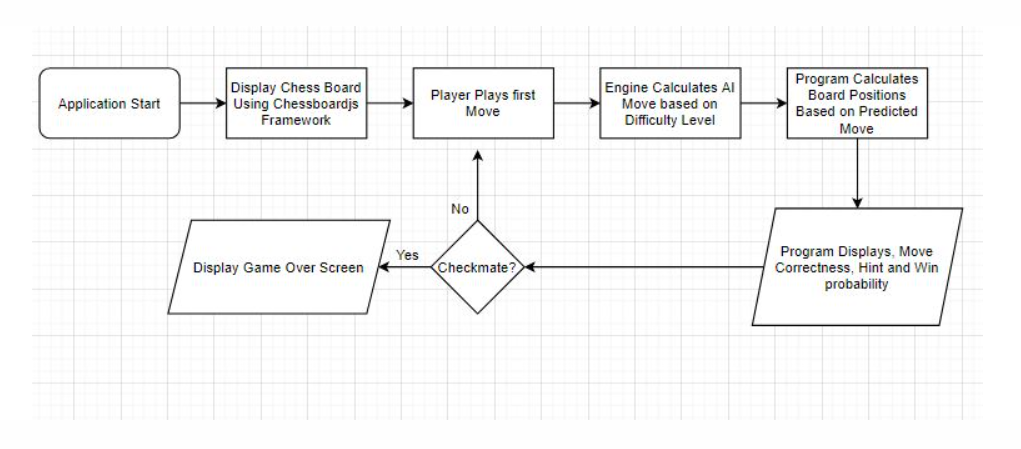
We have created a web app that allows the users to play against a bot and analyse their every move to get better.

**Solution Description**

We use the Stockfish Chess engine to provide the user with a bot with difficulty levels ranging from a 1200 rating bot to a Grandmaster Level bot.

The Web App shows various information regarding the game as it progresses like the current winning probability, Mistakes, Inaccuracies, Blunders and Also an option to get the best possible move at the given Board State.

**Architecture Diagram**

****

**Technical Description**

**Frontend**

The frontend of this application is built with HTML, CSS and JavaScript. Integrating seamlessly with a very popular framework for drawing Chess boards, Chessboardjs. (<http://chessboardjs.com/>)

We have embedded this tool into our app for drawing the frontend and keeping track of all the pieces on the board easily.

This HTML page is served with the help of flask framework on the backend.

**Backend**

For the backend, Flask was our best option as it allowed us to integrate a really powerful chess engine that is Stockfish, using the python stockfish library that makes it really easy to communicate with the stockfish engine executable. The Python Chess library was a cherry on top as it allowed us to manipulate the board and convert it into a FEN(Forsyth–Edwards Notation ), which is essentially an entire chessboard converted into a string. We can then evaluate the moves made by the user as well as predict the move that the bot should play.

As the Player starts playing against the bot of their desired difficulty level, the chess engine simultaneously calculates the best move for the BOT as well as producing a best case scenario for the human player in the form of a suggested move.

This calculation is based on a score known as the Centipawn notation which gives different values to pieces based on their position, threats and advantages. According to this notation,

* + - * The queen is worth 900,
      * Each rook is worth 500
      * Each knight is worth 300
      * Each bishop is worth 300
      * Each pawn is worth 100 centipawns.

And since the king is a key piece which cannot be captured or traded, it is assigned an arbitrarily large value.

The approximate winning percentage can be calculated using the centipawn advantage using the following Mathematical formula:

After this is calculated we can also determine the quality of the move based on the centipawn increase/decrease.

An inaccuracy is a decrease of 40 centipawns,

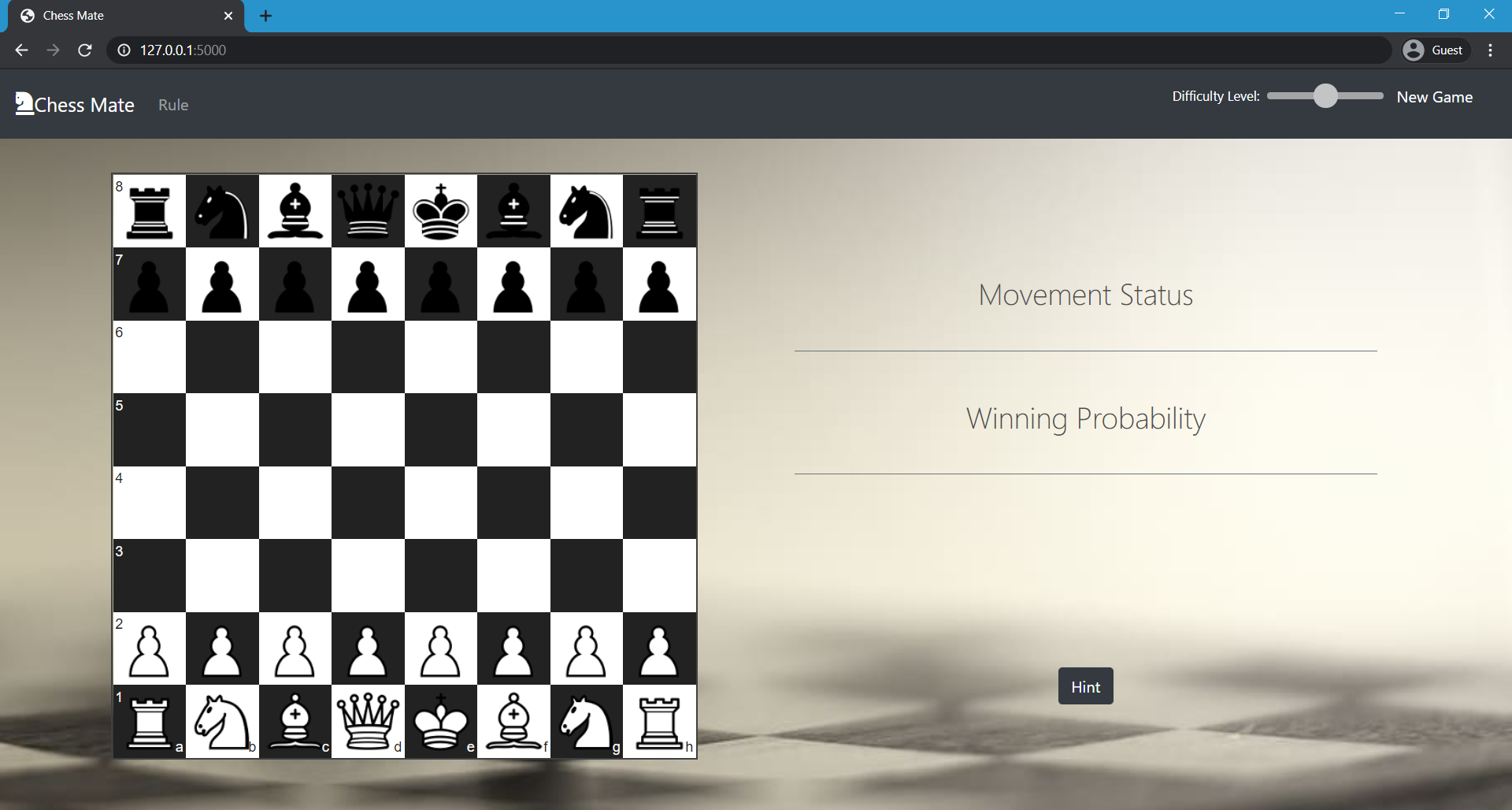
A mistake is denoted by the loss of 100 centipawns

and A blunder is a loss of 200 centipawns or more.

Any positive increase in centipawns is considered as a good move.

After all this information is calculated it is then communicated through the flask framework to the frontend JavaScript and then displayed on the Page for the user to disseminate.

**Screenshots**



**Live demo :**

<https://www.chess---mate.herokuapp.com>

**References**

* <https://chessboardjs.com/> - for drawing the board
* <https://pypi.org/project/stockfish/> - for calculating and evaluating moves
* <https://pypi.org/project/chess/> - for Board Parsing and manipulation
* <https://www.chessprogramming.org/Pawn_Advantage,_Win_Percentage,_and_Elo> - for providing very useful insights on calculating Win probability and Pawn Advantage