**Problem Statement:**

The project is to develop a ‘**Checkers**’ board game having user friendly and smooth GUI.

**Introduction**

Checkers also known as draughts, is a group of strategy board games for two players which involve diagonal moves of uniform game pieces and mandatory captures by jumping over opponent pieces. Checkers is developed from alquerque. The term "checkers" derives from the checkered board which the game is played on, whereas "draughts" derives from the verb "to draw" or "to move".

The core objective of the application is to provide an entertaining experience to user with the functionalities like:

* Two human players will play against each other
* Decision by the AI or computer for a valid move
* Computer will decide and announce winner of the game according to the rules of Checker Game.

**Rules of the Game:**

* One player has dark pieces (usually black); the other has light pieces (usually white or red).
* Normal Pieces can only move forward diagonally
* Kings can move forwards or backwards diagonally
* Once a piece reaches the end of the board it becomes a king.

**For more information and rules please refer:**

**https://en.wikipedia.org/wiki/Checkers**

**Targeted Audience**

The application is designed and suitable for every individual who like the Game **Checkers** especially the young generation i.e., teenagers and kids.

**What is ‘successes’ meant for us?**

Our aim is to provide the user with a smooth and entertaining experience of checkers game and to provide clear instructions for the game and make it user friendly as well beginner friendly.

**Functional Specification**

* The user gets the option to select the size of board for playing checkers.

For example, the user gets the option for size selection like (8X8) or (10X10) to begin the game.

* The user gets undo and redo buttons to manage their moves. There will be specific constraints on the number of times the user can use undo (or) redo. If the user plays with other users, then he/she should get the permission from every other player before using undo (or) redo. The user can only use either of these options only once during the whole game.

The user will not get these options if he/she is playing with computer as their opponent.

To perform the above-mentioned undo (or) redo operations, we need to keep track of the previous steps the user has taken to reach the current position so that he/she can edit their moves. Here the co-ordinates of the previous steps stored in the database in JSON format will be used.

* The users will get the option to save their game at the intermediate state they were playing at and the option to resume their previously saved game at any point of time and continue the game where they left.

While the user is playing the game against another user, he/she should get the permission of the other user to the save the game.

* The user can view their game history which provides basic statics like which one of the games user has won (or) lost, the accuracy of moves used, the winning percentage etc.
* The user needs to keep the track of time which he/she takes before every move. If the user is not able to complete his/her move in the assigned time period, their turn will end.
* The user also needs to consider the possibility of validity of the move they will use. If the move used is invalid then a pop-up will be displayed showing the warning message informing the user about the invalid move used by them.
* Considering the level of user, if the user is a beginner, then upon clicking on the piece they want to move, they will be able to see the possible moves they can attempt.
* The user will get the option of selecting the colour of pieces on the board as well as the colour of the board of their liking.
* Rather than user selecting their own level in human vs AI mode, the game AI will detect the level of user by considering the accuracy of the moves used by the user and then adjust its difficulty level accordingly. The AI will start playing from easy difficulty and then raise its level according to its own judgement of the user. The user will also be shown his/her own level after every game.
* The point system will also vary according to performance of the user. If the King is crowned and the user wins the game, then the points earned by the user will differ from the situation where the user wins without the King being crowned.
* In the Phase-II of the game, the user will get an option to play the game online with other players on a competitive level. The user will get an option to create a room to play will other users in multiplayer mode, where he/she can get a unique code so that they can play the game with some specific users like friends and family.

**Basic System Requirements for User**

* **Minimum hardware requirements**

Processor: intel core i3 7th gen

Memory: 2 GB RAM

Graphics: Integrated graphics card

Storage: 1 GB available space

O.S: Windows 7 -32bit

* **Recommended hardware requirements**

Processor: intel core i3 10th gen

Memory: 4 GB RAM

Graphics: 1 GB VRAM

Storage: 1 GB available space

O.S: Windows 8 64bit

**External Interface Requirements**

**Communication Protocols**: As of now, we have not planned any communication protocols for this project. As we will be going ahead with the project, we will be researching about the same and if we find anything that could make our project more efficient, we will be adding it.

**File Formats:**

**Json:** It is used for storing and transferring of data between the browser and the server.

.**py:** It contains the python source code. It can be edited with the help of any editor.

**.exe:** Contains the encoded instructions that the system can execute directly.

**Supported Devices**: This software is developed for Windows 32-bit or 64-bit.

**TECHNICAL SPECIFICATIONS**

* **HARDWARE Requirements**

RAM-4GB

CPU/Processor > intel i3

* **OPERATING SYSTEM**-windows 7 or greater
* **PROGRAMMING LANGUAGE -**Python >v 3.1
* **Database** – MongoDB
* **Framework** – Tkinter
* **IDE :** Vs Code > v1.31

**Future Scope/Extensions**

In human vs AI mode, we have Different difficulty Levels of game i.e. Easy, Medium, Hard

**Network Checkers.**

Two players will play over local networks