**PROJECT PROPOSAL**

*Poker*

Table of Contents

[Project Abstract 3](#_Toc113300636)

[Conceptual Design 3](#_Toc113300637)

[Proof of Concept 3](#_Toc113300638)

[Background 3](#_Toc113300639)

[Required Resources 3](#_Toc113300640)

## 

## Project Abstract

*This project will be an implementation of an internationally popular card game,* ***Poker****. The game will feature two modes:* ***Texas Hold ‘Em*** *and* ***Blackjack****. The game will be hosted on a website. Upon entering the website, the user will be welcomed with a satisfyingly designed interface. The user will then have the option to choose from one of the two game mode’s as well as if they want to play against an artificial intelligence inspired computer player or a friend on another computer device.* *Lastly, the user will enter their ‘nickname’ for the duration of their playtime.*

*During actual gameplay, the cards will be physically visible on the screen, along with a pleasant color palate and design. A feature that will be a bonus if its execution is possible is to keep a leaderboard of the best players, where the top players will have their nicknames listed with their win count.*

## Conceptual Design

*Firstly, the front-end will be built with HTML & CSS. For the back-end functionality of this project, a programming language such as Go or Python will be used. For the latter, the web development framework known as Flask will manage, as it is very portable and well-documented. Both Flask and Go will enable us to render templates and easily include the rendered card photos. If my team and I are hopeful about including a leaderboard, it will require a small database to host the usernames and win counts. For this, a simple MongoDB or SQLite database would suffice and would most likely need to be hosted remotely or temporarily hosted with the running of the game.*

*Playing another user on a separate machine will require knowledge of networking and socket communications. This feature will be a bonus, which will be replaced solely by only playing against the computer if the former is not feasible given the timeline of the project at that moment.*

## Proof of Concept

<https://github.com/briangunsel/gamBIT>

## Background

## Required Resources

* Database (SQLite, MongoDB)
* Web hosting
* Networking knowledge (multi-machine playability)
* Python (Flask) or Golang
* 52 images of cards in a standard playing deck