Design:

Introduction:

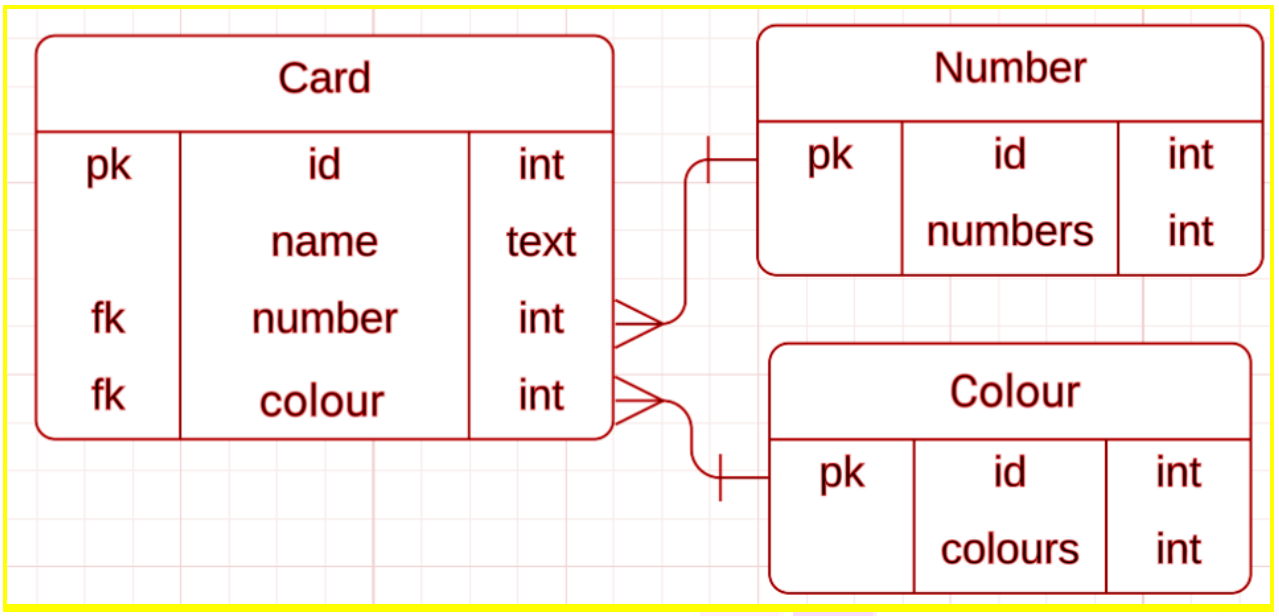
The project I am going to do in year 12 is a website design to allow people to play poker. This should be on a website, allowing people to access. To achieve this, I will have to use Python, Flask, Html5, Css, Database as these languages can help the making of the project.

Design:

The whole project should include instructions and an enhanced playing experiment. This will meet the requirement of the user which are poker players. The colour, fonts, and experiment should be as close to reality as possible. I will use #166A14, #249450 as the main colour of the project. To achieve this I will follow the following plannings.

Database design and ER diagram:

As the database, the following ERD diagram is my planning of the structure.



It should contain 52 poker cards (excluding jokers). And contain the colour and number of the cards.

The sqlite query will be
“select Card.id,Colour.colours,Number.numbers,Card.name from Card
join Colour on Card.colour = Colour.id
join Number on Card.number = Number.id
where Card.id = {ran}
order by Card.id”
({ran} is a variable in the python file)
This will be in the python code.
Because this is not exactly an information page, the result will not be shown directly to the user, it will only be calculated in the program and show images at the end.

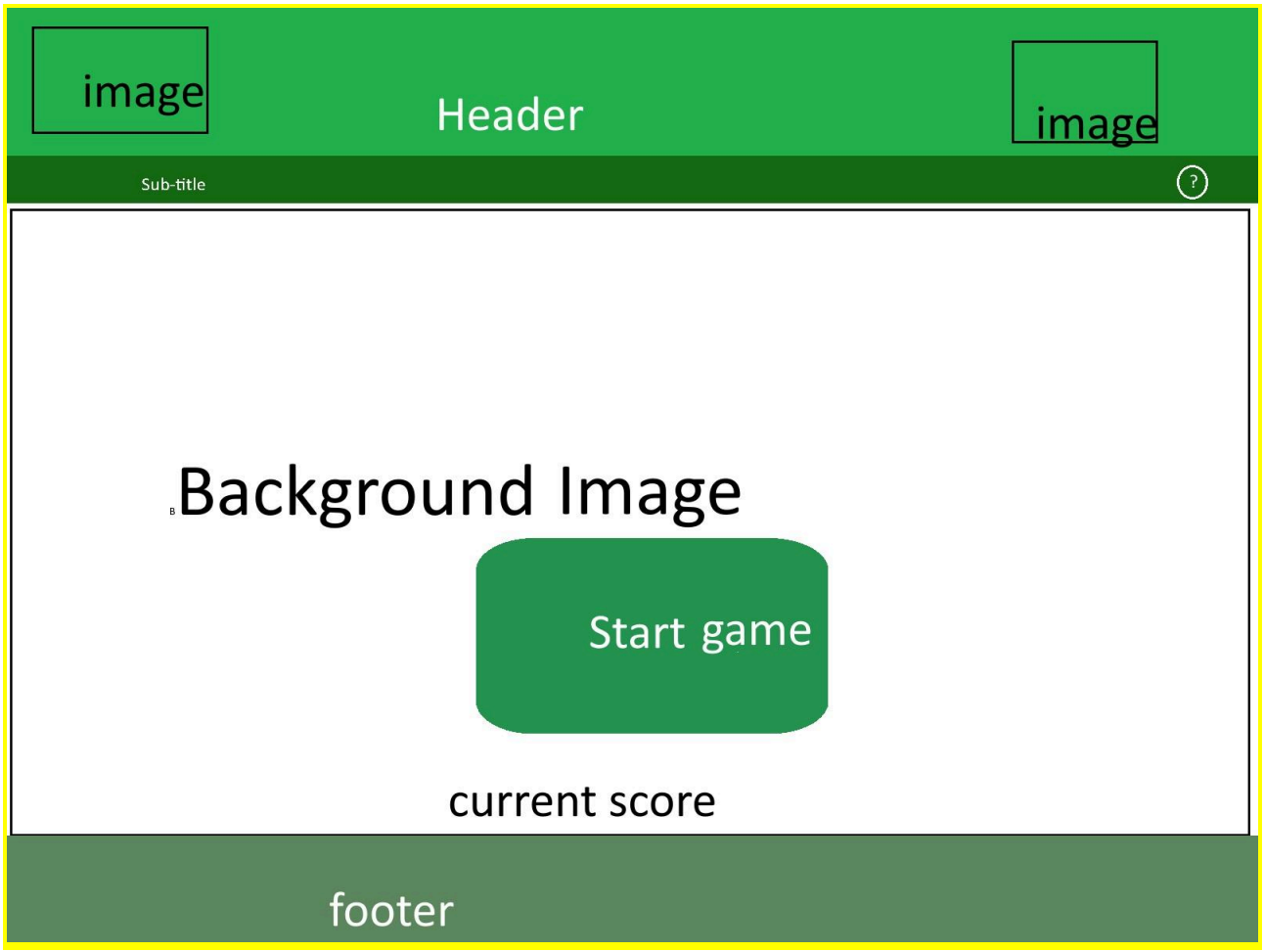
Website designing:

For the website design, I will use calibri as the font, and #166A14, #249450 as the main background colour. The colour represents the poker playing table and brings the user better experience.

The website will contain a main page and a playing page, the main page will use the @app.route(“/”) as the Flask route, and the playing page will use @app.route(“/play”) as the Flask route.

The main page of the website will have the following structures.
- The poker background image.
- Clear button to play.
- Instructions to new players.
- The current money.

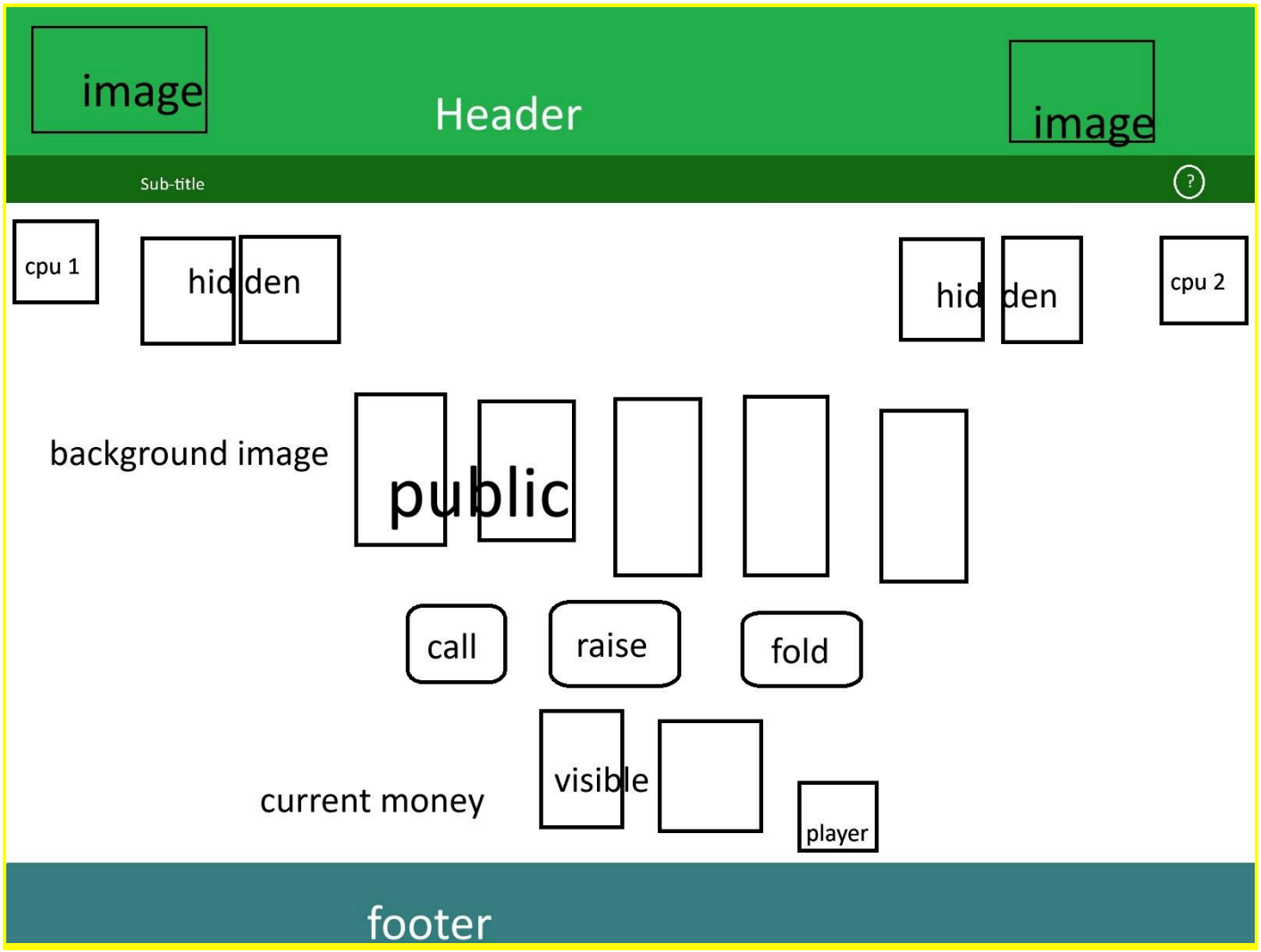
The following image is the approximate design of the page.



The playing page of the website will have the following structures.

- The green background colour.
- The clear, visible public cards allow players to see.
- The significant hand cards only the player can see.
- The hidden cards of computer players.
- The clear buttons to call, raise, fold.
- The clear button to exit to the main page.

The following image is the approximate design of the page.



TECHQUITY

Documentation:

Web Application Development Log

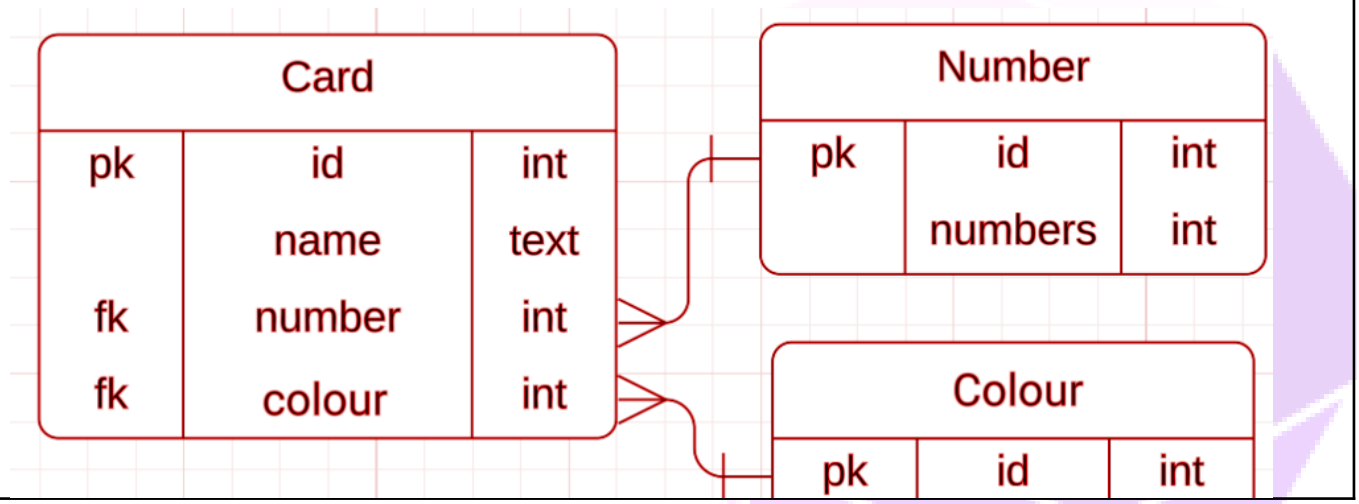
Design Section

Project link on Github:
https://github.com/jasper22624/12-DTP

Project name:
Poker playing website

Project Summary: Include purpose and target audience
The purpose of this project is to develop a poker game website that provides players with an interactive and engaging online experience. The target audience is poker players, both casual and experienced, who want to enjoy a realistic card-playing environment in their browser. The website will include instructions, a card database, and gameplay features to ensure the outcome is both functional and entertaining. Colours, fonts, and user interface elements are designed to closely match real-world poker to enhance the playing experience.

Initial Database Design ERD:



The initial database design includes three tables, they contain 52 cards and information about the cards used for calculations.

WebPage Design- Wireframe and/or Palette

The main webpage design includes:

A homepage with a “Click to Play” button leading to the game table.

A poker table layout with player areas, cards, and chip counts.

A simple navigation menu with “Home” “How to Play”

Colours: green background (poker table feel), white text, and red/black suits for cards.

Relevant Implications (explain at least 3)- Before Development.

The three important implications about my project are usability, functionality, ethical, and legal.

The site must be easy to use for all players. I used a clear layout with a poker table theme, readable fonts, and colours that match real cards. Buttons like “Play” are obvious, and the design is modern to make it look better.

The game must run correctly. Cards are dealt without duplicates, winners are calculated accurately, and chip counts update properly. Testing checked links, game logic to make sure everything works reliably.

I avoided copyright issues by using only open-licensed or original assets. I also made the game for fun, not real gambling, so there are no issues with betting laws. All sources were credited correctly.

For the ethical side, I also considered ethics by using a variable to represent money instead of real currency, making the game safe for entertainment.

Sprint #1

Sprint #1 Goals **(before starting sprint 1):**
The main webpage design includes: A homepage with a “Click to Play” button leading to the game table. A poker table layout with player areas, cards, and money counts. A simple navigation menu with “Home,” “How to Play,” and “About.” Colours: green background (poker table feel), white text.

Sprint #1 Testing

Include tests for everything in your application including all links, input forms, correctness of data and correctness of queries. Add more columns as required.

Testing Table			
What are you testing	**How are you testing it**	**Expected Result**	**Pass/Fail**
Extract all cards details	Using SQLite query	ID, colour, number of cards showed correctly	Pass
Extract specific card detail	Using SQLite query	ID, colour, number of card showed correctly	Pass
Extract random card detail	Using SQLite query	ID, colour, number of card showed correct	Pass

Sprint #1 Review

Progress Review (100+ words)

In Sprint 1, I focused on building the base of my project and making sure everything was set up properly before adding more advanced parts. I started by creating a GitHub repository, which became the main place to store and manage my code. I made regular commits to track my progress, and this gave me a clear record of what I had done. If I made a mistake, I knew I could look back or undo changes. Setting this up early helped me stay organised and follow good practice.

After that, I set up Flask as the framework for my website. At first it was a little tricky, but after testing I managed to launch it with a simple homepage. Even though the page was basic, it was an important step because it showed that Flask was installed correctly and working. Having a homepage also gave me something I could see and test, which made the project feel real and ready to grow in the next sprints.

Next, I built the database, which was very important for the game. I added all 52 playing cards with the right suits and numbers. To make sure everything worked, I tested the database by running different SQL queries and checked that the results were correct. At first, some of the queries did not give me the answers I wanted, but after trying out new SQL statements and fixing the structure, the problem was solved.

Overall, Sprint 1 was successful. By setting up GitHub, Flask, and the database, I built a strong foundation. This meant that in future sprints, I could spend more time adding gameplay and user features without worrying about setup problems.

Feedback	How I plan to address the feedback in next sprint
## Player 1 (programmer, Eric L): “I really like how the layout of your game is set up. The poker table theme makes it feel realistic, and the Play button was big and clear so I instantly knew what to do. The use of green background and white text fits the theme really well. One thing you could maybe add is a short welcome message or some quick instructions on the starting page, just so first-time players understand what’s going on before they press Play.”	I agree with this. My focus was on making the layout clear and simple, but adding a welcome or short guide could make the site even easier for beginners. It also makes the projects seem friendly and improves social connection to users.
## Player 2 (programmer, Eric L): “The card dealing and money system are working really well. I didn’t see any duplicate cards when I tested a few rounds, and the chips updated every time correctly. This makes it feel fair and consistent. One improvement could be adding some kind of small animation when the chips go up	I’m glad this was noticed because the database and fairness of dealing were a big part of the project. I like the suggestion about animations — it would make the game more fun but I cannot achieve this suggestion currently due to the html and css limitations.

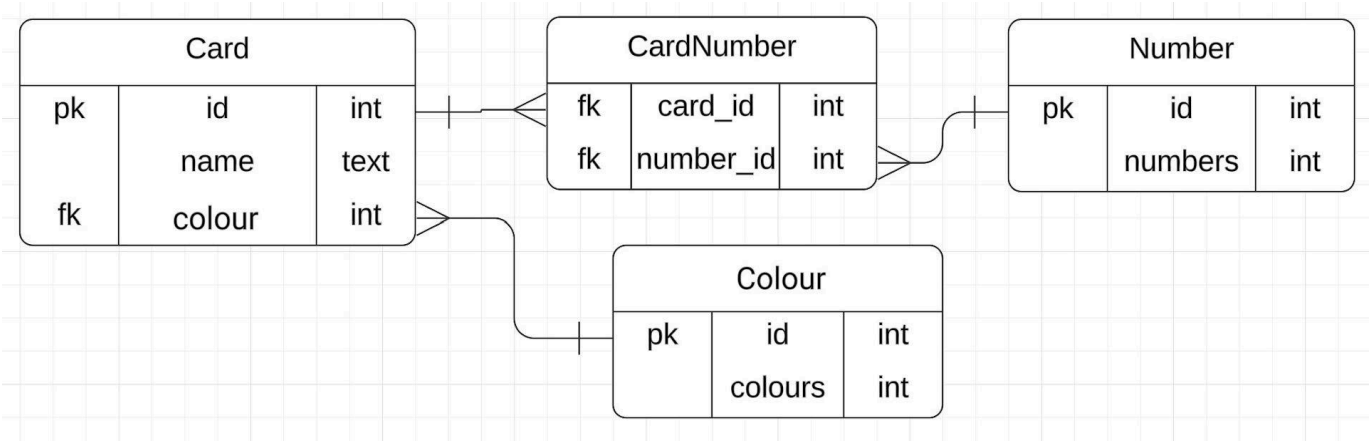
or down, because right now the numbers just change instantly and it could feel more engaging if it was shown visually.”	

Sprint #2

Sprint #2 Goals **(before starting sprint):**
The webpage design includes: A playing page with basic gameplays and correct calculation. A poker table layout with status, winning information, and buttons. Better colouring choices to simulate real world feeling. Add texts to make the player feel welcomed. Colours: green background (poker table feel), white text, desaturated buttons.

Database improvements (if any)

Entity Relationship Diagram



The added table makes the database more complex and deals with a specific card Ace. It now can act as number 1 when forming straight.

WebPage Design Improvements (if any)- Wireframe/Sketches/Pallette/Font etc

Added poker table style background.

Improved readability by adjusting colours.

Show cards as images rather than text — this makes the game more engaging.

Make winning/losing messages more visible.

Show the player's money clearly at the bottom of the screen.

Sprint #2 Testing

Include tests for everything in your application including all links, input forms, correctness of data and correctness of queries. Add more columns as required.

Testing Table			
What are you testing	**How are you testing it**	**Expected Result**	**Pass**

			/Fail
Test a game on website	Check the results	Correct winning results	P
Test instruction page	Access on website	Correct display	P
Test button that link to other page	Access on website	Correct links	P
Test basic money calculation on python	Run the program on VSCode	Correct money calculations	P
Correct image showing of cards	Inspect images on browser	Correct images	P

Sprint #2 Review

Progress Review (100+ words)
The webpage design was made to give players a realistic poker experience, and I think it does this well. The main playing page includes all of the basic gameplay features, such as dealing cards, placing bets, and showing the results of each round. The calculations for winning are correct, which is very important because it makes sure the game is fair and reliable. This means that the core parts of the poker game work as expected and give users a proper experience when they play. The layout of the poker table is clear and easy to understand. It shows the player's status, how much money they have, and who won each round. There are also interactive buttons, such as "Play" and "Fold," which make it simple for the user to control the game. Because everything is in the right place, the player does not get confused and can follow the game flow easily. The colour choices were also improved to make the game feel more like real-world poker. The green background gives the sense of a proper poker table, while the white text makes sure that the information is easy to read. The buttons use more desaturated colours, which means they do not distract from the cards or the table but still look neat and professional. These choices give the design a stronger theme and help create the right atmosphere for the game. Overall, the design is not only functional but also visually clear. It matches the intended poker theme, makes the game simple to play, and provides a professional look that adds to the player's experience.

Feedback	How I plan to address the feedback in next sprint

Player 3 (end user, William R): “I don’t know how the coding works, but from just playing it, the game feels fun and easy to use. The table design looks like a real poker table, and I could follow the flow of the game without being confused. The buttons were big enough and clear, which made it simple to play. Maybe you could add some visual effects when clicking the buttons, like a slight animation or colour change, because that would make the game feel more interactive for players like me.”	I’m happy they could enjoy the game without needing to understand the logic behind it, because that shows the design is user-friendly. I also like the suggestion about button effects — it would make the interface feel more engaging and responsive. I will make the button smaller when clicking it, it makes the feeling of clicking the button in real life.

TECHQUITY

Sprint #3

Sprint #3 Goals

(before starting sprint):

The webpage design includes:

A playing page enhances gameplays and corrects money calculation.

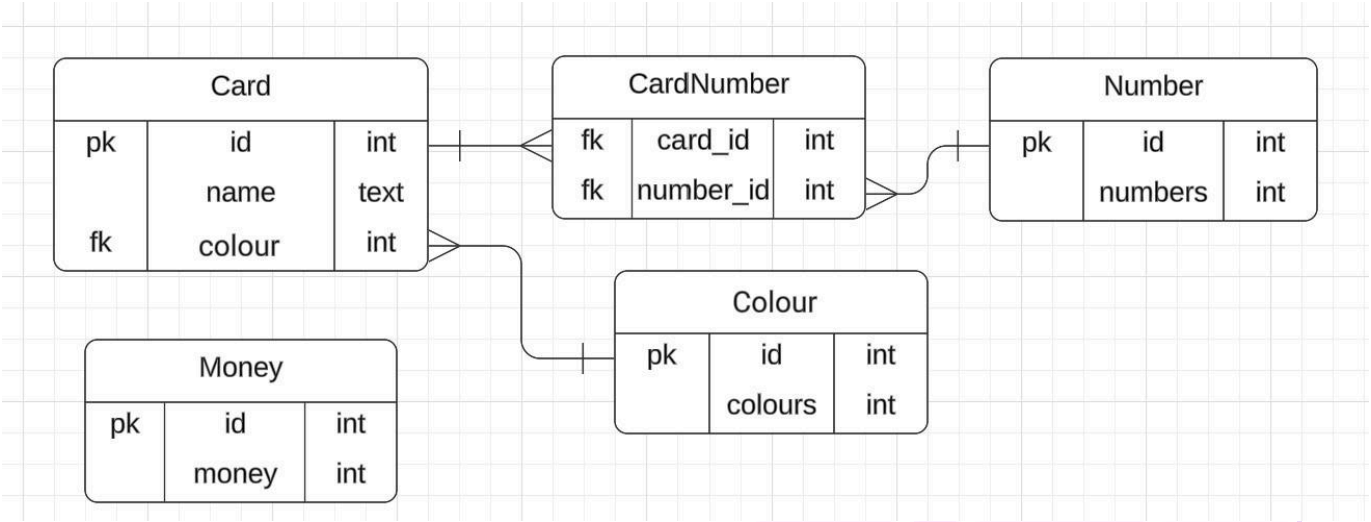
A poker table layout with status, winning information, and money showing.

Add hover effect to buttons and make the visual effect when pressing.

Better instruction page to help new players learn how to play.

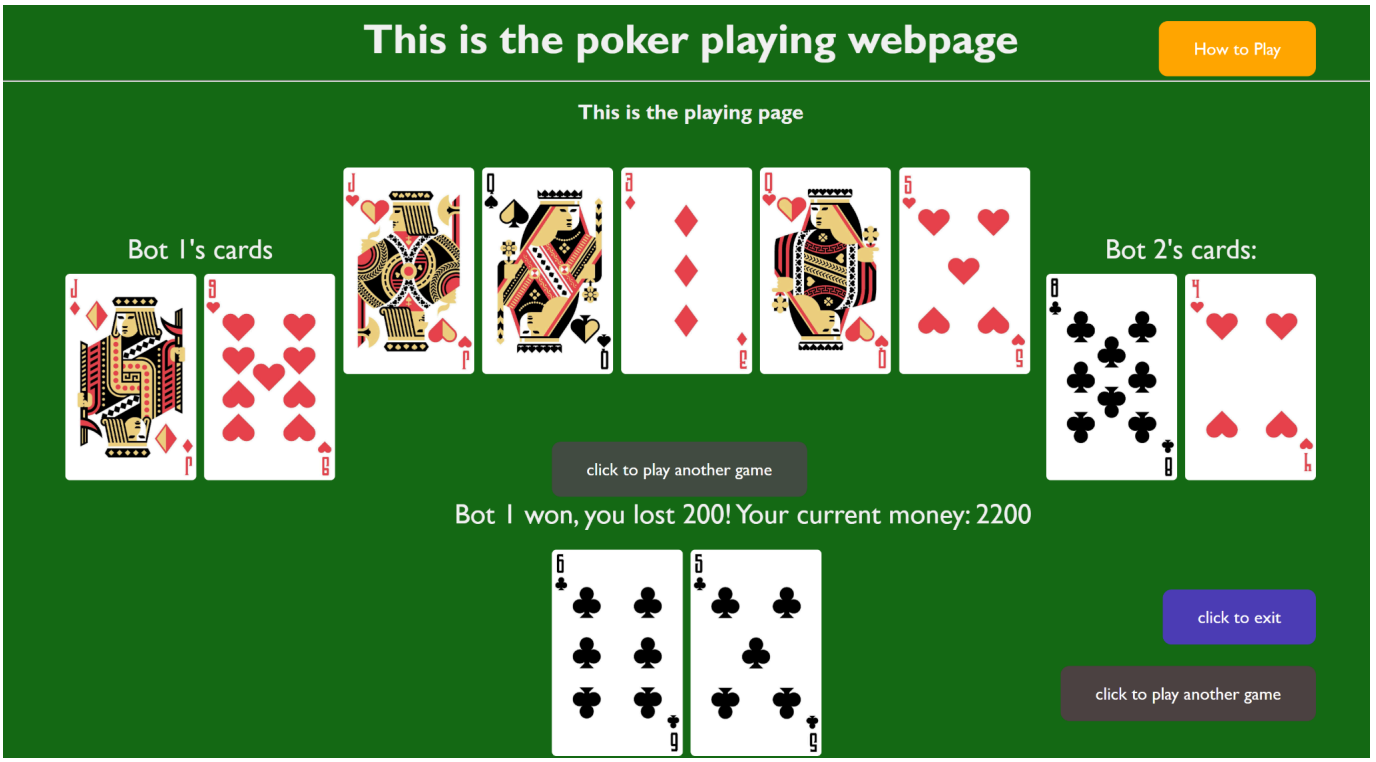
Database improvements (if any)

Entity Relationship Diagram



The added table can provide storage for current money the player got, it is initialized at the start of the program.
I used queries to update and extract from the Money table.

WebPage Design Improvements (if any)- Wireframe/Sketches/Pallette/Font etc



The enhanced playing page can show players with a real world simulating experience. Including correct calculations and variable showing. It contains links to other pages and clear instructions to users.

Relevant conventions:

During development I applied Nielsen's usability heuristics to guide improvements in the design of my poker website.

The first heuristic I focused on was Consistency and Standards. In early versions, the buttons looked different across pages and sometimes text sizes changed. I refined this by using the same button style (rounded corners, green fill, white text) across all pages, and keeping the same font for headings and instructions. This consistency helped the site feel professional and made it easier for users to recognise actions quickly.

I also used the heuristic of Visibility of System Status. Originally, the game only showed "You won" or "You lost" without updating the chip count until the next round. I changed this so that the money updates straight away and the win/loss message is shown in bold. This gave players immediate feedback, reduced confusion, and made the game feel responsive.

Finally, I applied for Error Prevention. A major risk in poker is drawing duplicate cards, which would break the game. To prevent this, I set up checks in the code so that a card cannot be drawn twice. I also tested this across multiple rounds to confirm it worked. This made the gameplay more reliable and kept the database accurate.

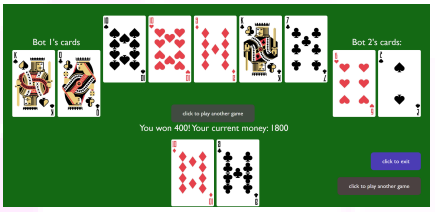
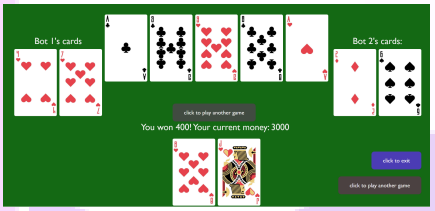
By applying these heuristics, I improved the usability of the website and made the final outcome clearer, more consistent, and more reliable for players.

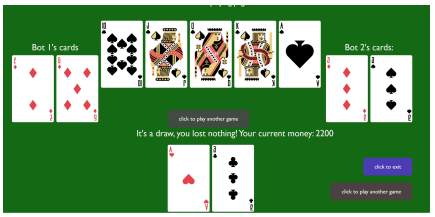
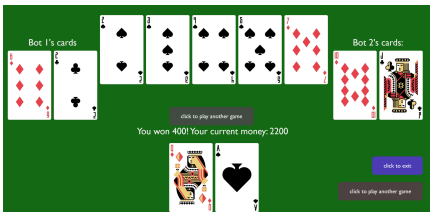
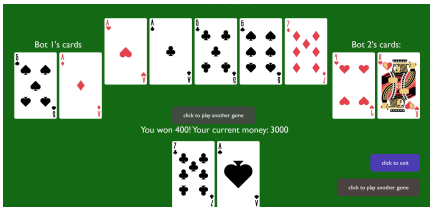
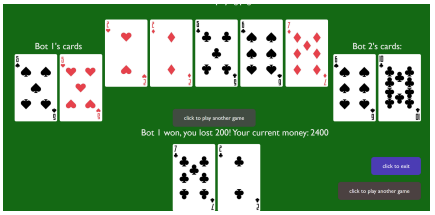
Sprint #3 Testing

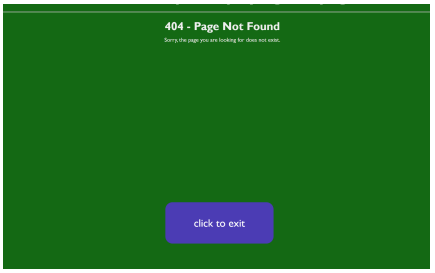
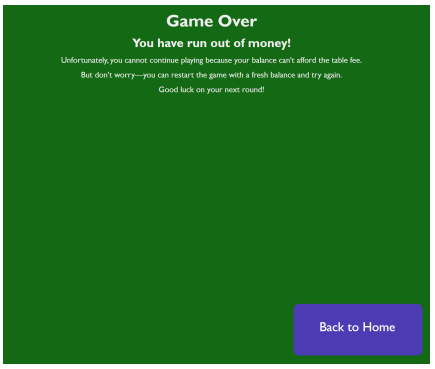
Include tests for everything in your application including all links, input forms, correctness of data and correctness of queries. Add more columns as required.

Testing Table			
What are you testing	**How are you testing it**	**Expected Result**	**Pass /Fail**

Same card won't be extract twice	Repeat game 100 times, refine logic in app.py	No repeat cards, correct python logic
		`random_card = random.randint(1, 52)`
		`while selected_cards.count(random_card) >= 1:`
		`  random_card = random.randint(1, 52)`
			P

Money is tracked correctly	Test game in win, lost, draw, and fold	Calculation is correct It's a draw, you lost nothing! Your current money: 800 Bot 2 won!, you lost 200! Your current money: 1600 You won 400! Your current money: 2000 You folded! Your current money: 1900	P
Game result calculate correctly	Test game in win, lost, draw, and fold	Calculation is correct  Fullhouse greater than two pairs  Even the pairs are same, J is greater	P

		than 8  This is a rare situation, where all player's cards are same, the calculation is correct  This shows the program calculated 1, 2, 3, 4, 5 correctly, not forcing Ace be 14  This shows the full house with same threes and different twos, the calculation is correct  The threes in full house must be considered firstly, 5>2, the calculation is correct	
Card image is same as in calculation	Use inspect tool in browser, print in .py	Shows the same as in calculation, this makes sure the data integrity of the project	P
Card image is the actual card	Use print in .py, check database, check /static	Show the same card name, card image in the database, this makes sure the data integrity of the project	P

		1 Diamond 2 1 images/Playing-Cards-Jumbo_41.jpg 2 Diamond 3 1 images/Playing-Cards-Jumbo_42.jpg 3 Diamond 4 1 images/Playing-Cards-Jumbo_43.jpg 4 Diamond 5 1 images/Playing-Cards-Jumbo_44.jpg 5 Diamond 6 1 images/Playing-Cards-Jumbo_45.jpg 6 Diamond 7 1 images/Playing-Cards-Jumbo_46.jpg 7 Diamond 8 1 images/Playing-Cards-Jumbo_47.jpg 8 Diamond 9 1 images/Playing-Cards-Jumbo_48.jpg 9 Diamond 10 1 images/Playing-Cards-Jumbo_49.jpg 10 Diamond J 1 images/Playing-Cards-Jumbo_50.jpg 11 Diamond Q 1 images/Playing-Cards-Jumbo_51.jpg 12 Diamond K 1 images/Playing-Cards-Jumbo_52.jpg 13 Diamond A 1 images/Playing-Cards-Jumbo_40.jpg 14 Clubs 2 2 images/Playing-Cards-Jumbo_02.jpg 15 Clubs 3 2 images/Playing-Cards-Jumbo_03.jpg 16 Clubs 4 2 images/Playing-Cards-Jumbo_04.jpg	
If it jumps to 404 page when incorrect URL	Enter wrong URL	Jump to 404.html 	P
Game stops when money<0	Fold until money<0	Jump to restart.html and restart whole game 	P

Feedback	How I plan to address the feedback
Player 4 (end user, David D): “Maybe you could use brighter colours for the buttons to make them pop more. It might catch the eye quicker.”	I decided not to do this because I wanted the colours to look natural, like a real poker table, rather than too flashy. I think keeping the buttons more muted fits the atmosphere of poker better.
Player 5 (end user, Elissa P): “For some players that are new to poker, the instruction page is not clear enough to understand how poker works. I suggest the instruction page be clearer and more friendly to new players”	This is an important factor about the project, it reminds me the layout is not the only thing in displaying, but also the texts. I will improve my instructions and other texts, to improve the overall quality of the project.

Final Review

What were the results of your testing? Did you get through all that you expected to? What is causing problems? What went well? What was a challenge? What would you do differently if you had a chance?

Discuss how planning, testing and trialling led to the development of a high-quality digital technologies outcome.
Planning, testing, and trialling were important for creating a high-quality outcome. By planning sprints with clear goals, I was able to stay organised and focus on one task at a time. Testing confirmed that features such as card dealing, chip updates, and queries worked correctly. Trialling different layouts and functions helped me find the most effective solutions. These steps meant errors were fixed quickly, improvements were continuous, and the final poker website became reliable, user-friendly, and close to the intended design. In my poker website, planning was an important step because it helped me structure the database and set up the main features in a logical way. I planned how the cards, players, and games would be stored in the database so that the program could use them easily. This meant that the card dealing system would work without errors, and the logic behind the game was clear from the start. Planning also made sure I had a strong foundation to build on, so later features such as betting and winning could be added smoothly. Testing played a big role in making sure everything worked correctly. I tested the dealing of cards to confirm that no duplicates appeared. I also tested the winner calculation to make sure the right hand was chosen at the end of each round. Another part of testing was checking that chip counts updated properly when a player won or lost. By going through different test cases and checking the database updates, I was able to make sure the site stayed reliable. Trialling design choices also improved the overall user experience. I tried out different colour schemes, button styles, and layouts to see which options looked and worked best. For example, the green background and desaturated buttons made the game look more like a real poker table, while still being easy to read. By trialling these design options, I could see what worked well and what did not, and then keep the best ideas. Together, planning, testing, and trialling helped me create a website that was both functional and easy to use. These steps gave the project structure, made it reliable, and improved the user experience.

How I addressed the Relevant Implications

As discussed before the project, what relevant implications had you identified at the beginning and how did you manage to address them as you went. Were there any others that you came across while making this game? How did you manage to address them? 100+ Words

At the start of the project, I identified usability, functionality, ethical, and legal implications.

Usability means making a system easy to understand and navigate. In a database-driven project, poor usability can make it hard for users to find or interact with data. I addressed this by designing a clear poker table layout, readable fonts, and responsive pages so players could access information easily.

Functionality refers to the database working correctly and returning accurate results. A database that produces duplicate or incorrect data would break the game. I addressed this by testing queries to ensure cards were unique, winners were calculated correctly, and chips updated as intended.

Ethical issues are about ensuring the system is responsible and fair. In databases, this can relate to how sensitive data is stored or how results are presented. I addressed this by representing money as a variable instead of real currency, ensuring the site is for entertainment only and does not promote gambling.

Legal implications include respecting copyright and data protection laws. Databases can easily include copyrighted content or store personal information. I addressed this by only using open-licensed or original assets and avoiding any collection of sensitive personal data.

By reflecting on these implications throughout development, I created a database-driven poker website that was functional, safe, and ethical.

TECHQUITY

Resources:

https://mondaymandala.com/printable-playing-cards/ Poker cards images

Marking Schedules: For Teacher use only

AS91892- Use advanced techniques to develop a database

Credits: 4 (internal)
NZQA: https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91892.pdf

Achieved- Develop a digital outcome to manage data	**Evidence**	✔
designing the structure of the data	Made an ERD	
using appropriate tools and techniques to structure, organise, query and present data for a purpose and end user	Competently normalized data structure Multiple queries across linked tables Basic Python CRUD functionality with text input/output	
applying appropriate data integrity and testing procedures	Testing Logs. Do the queries work and return what is expected?	
Explaining relevant implications.	Relevant Implications Questions answered appropriately	
Merit- Develop an informed digital outcome to manage data		
using information from testing procedures to improve the quality of the database	Suggested improvements implemented- improve database design, add or refine entities or tables and relationships Made improved ERD(s)	
structuring, organising and querying the data logically	No problems with structure and datatypes in the table. No wildcard queries, JOIN's used efficiently. 2NF at least.	
addressing relevant implications.	The outcome addresses the relevant implications mentioned	
Excellence- Develop a refined digital outcome to manage data		
iterative improvement throughout the design, development and testing process	Improved functionality of database and refinements to structure of the data.	

presenting the data effectively for the purpose and to meet end-user requirements.	Very refined outcome providing all needed functionality	

TECHQUITY

AS91893 - Use advanced techniques to develop a digital media outcome

Credits: 4 (Internal)
NZQA: https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91893.pdf

Achieved	**Evidence**	✔
using appropriate tools and techniques for the purpose and end users	Coded a website in HTML and CSS. Looks and functions mostly like intended in the design.	
applying appropriate data integrity and testing procedures	It functions as intended and the right pages show with the correct data when the links are clicked	
using relevant conventions for the media type	Site has basic usability (HCI)	
explaining relevant implications.	Relevant conventions section completed adequately	
Merit		
using information from testing procedures to improve the quality of the outcome	Testing and sprint reviews show evidence of specific improvement made based on the testing. No lorem ipsum anymore!! Accurate Data	
applying relevant conventions to improve the quality of the outcome	Decent HCI. Eg. generally shows “consistency and standards”, and/or “help and documentation” by having an about page.	
addressing relevant implications.	Has met relevant implications like NOT used copyright material, addresses privacy by keeping data safe (eg hashed passwords) etc.	
Excellence		
iterative improvement throughout the design, development and testing process to produce a high-quality outcome	Must be high quality. No obvious errors or inaccurate data.	
using efficient tools and techniques in the outcome’s production.	Kept a good log, used github to keep backups and record of work, lots of commits on github.	

TECHQUITY

AS91896 - Use advanced programming techniques to develop a computer program

Credits: 6 (Internal)
NZQA: https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91896.pdf

Achieved	**Evidence**	✔
writing code for a program that performs a specified task	Applications works as intended.	
using advanced techniques in a suitable programming language	“creating methods, functions, or procedures that use parameters and/or return values” flask requires that “responding to events generated by a graphical user interface (GUI) ”- links in webpage run routes that do queries “using additional non-core libraries”- using Flask.	
setting out the program code clearly and documenting the program with comments	Some code comments, code layed out relatively cleanly with constants/variable at the top, routes and functions after.	
testing and debugging the program to ensure that it works on a sample of expected cases.	Testing table shows the basic web page functionality was tested.	
Merit		
documenting the program with appropriate names and comments that describe code function and behaviour	Good variable and function names. Good and plentiful code comments.	
following common conventions for the chosen programming language	PEP8 Followed. Use Pylint or similar and check linting.	
testing and debugging the program effectively to ensure that it works on a sample of both expected cases and relevant boundary cases.	Lots of testing including request for non-existent id returning an error 404 page.	
Excellence		
ensuring that the program is a well-structured, logical response to the specified task	Enhanced program, e.g. functions that aren’t routes to make the code cleaner (eg query_db). Neat and logical code.	
making the program flexible and robust	Great structure making code easy to extend. Unbreakable code. Eg. Error-404 and error 505 handlers	
comprehensively testing and debugging the program.	Unbreakable code with a lot of tests done for every testable element of the program including unexpected input like a non existent route or page returning custom error 404 handler	

TECHQUITY

AS91897- Use advanced processes to develop a digital technologies outcome

Credits: 6 (internal)
NZQA: https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91897.pdf

Achieved	Evidence	✔
using appropriate project management tools and techniques to plan the development of a digital technologies outcome	Github used. Goals defined. Good commit messages.	
decomposing the outcome into smaller components	Evidence by regular commits with good messages describing the completed task	
trialing the components of the digital technologies outcome	Completed tasks committed should have been tested	
testing that the digital technologies outcome functions as intended	The whole application should be tested between sprints	
explaining relevant implications.	Relevant implications section complete	
Merit		
effectively using project management and version control tools and techniques to manage the development of a digital technologies outcome	Plenty well named commits done regularly (eg not a rush at the end or large period of inactivity seen in git log)	
trialing multiple components and/or techniques and selecting those which are most suitable	As above with larger number of small components tested and committed	
using information appropriately from testing and trialling to improve the functionality of the digital technologies outcome	Specific examples given in end of sprint reviews showing how they found something out in testing and improved their outcome as a result. Can be observed too.	
addressing relevant implications.	Application meets all relevant implications. Watch for non-copyright material, offensive material, poor functionality or aesthetics despite these issues being mentioned in testing.	
Excellence		
discussing how the information from planning, testing and trialling of components assisted in the development of a high-quality outcome.	There is a section at the end specifically asking this question.	