



BLACKJACK .NET

Team 3 - Final Project

Members:

- Aiden Macdonald
- Alexis Cooper
- Emanuel Quintana
- Haoxuan Ding
- Zach Jordan



PROJECT DESCRIPTION

Welcome to Blackjack.Net, the ultimate training grounds for mastering the art of blackjack without the risk of losing any money! Are you ready to elevate your blackjack skills to the next level?

At Blackjack.Net, we provide a safe and exciting environment for players of all levels to hone their strategies and tactics without the worry of financial losses. Whether you're a seasoned pro or a complete beginner, our platform offers an immersive and realistic blackjack experience that will challenge and inspire you to improve. Track your wins, losses, and ties as you play head-to-head against our casino-based dealer! Learn from your past performances, analyze your moves, and identify areas for improvement to boost your win percentage.






Say goodbye to the limitations of traditional casinos and say hello to unlimited practice and growth at Blackjack.Net. Sharpen your skills, refine your strategies, and become a blackjack master without ever risking a single cent. Get ready to play, learn, and win like never before at Blackjack.Net!



TOOLS IMPLEMENTED

	PROJECT Tracker (GITHUB PROJECT BOARD)	★ ★ ★ ☆ ☆	Useful in theory, but underutilized in our application
	VCS REPOSITORY (GITHUB)	★ ★ ★ ★ ☆	Progressed the sharing of code, added difficulties in merging ideas
	Database (POSTGRESQL)	★ ★ ★ ★ ★	Simplistic interface, and provide ample resources for our uses
	IDE (VSCODE)	★ ★ ★ ★ ★	A one-stop-shop for all our coding needs
	UI TOOLS (HTML/CSS, EJS)	★ ★ ★ ★ ☆	Difficult to get a grasp, but proved immensely helpful with dynamic gameplay

TOOLS IMPLEMENTED

	APPLICATION server <i>(NODEJS)</i>	★★★★☆	Slick, easy, and allowed for a more efficient workflow
	DEPLOYMENT Environment <i>(AZURE + DOCKER)</i>	★★★☆☆	Provided ample support for what we needed, but proved to be the most difficult to use
	COMMUNICATION <i>(DISCORD + ZOOM)</i>	★★★★★	Multiple mediums allowed for a diverse exploration of debugging and planning
	ADDITIONAL TOOLS <i>(DBDiagram.io + LUCIDCHARTS)</i>	★★★★☆	DBDiagram.io: Effective, but limited free tools Lucidcharts: Easy, Simple, Perfect
	PLANNING FOR THE FUTURE <i>(PHOTOSHOP, EXTERNAL APIS)</i>	☆☆☆☆?	Time permitting, we wanted to incorporate a custom background and external page depicting blackjack rules



CODING METHODOLOGIES

Initially, we aimed to embrace the Agile methodology for our project, but over time, we unintentionally veered towards a more ad-hoc approach, where tasks were implemented as needed without a strict adherence to Agile principles.

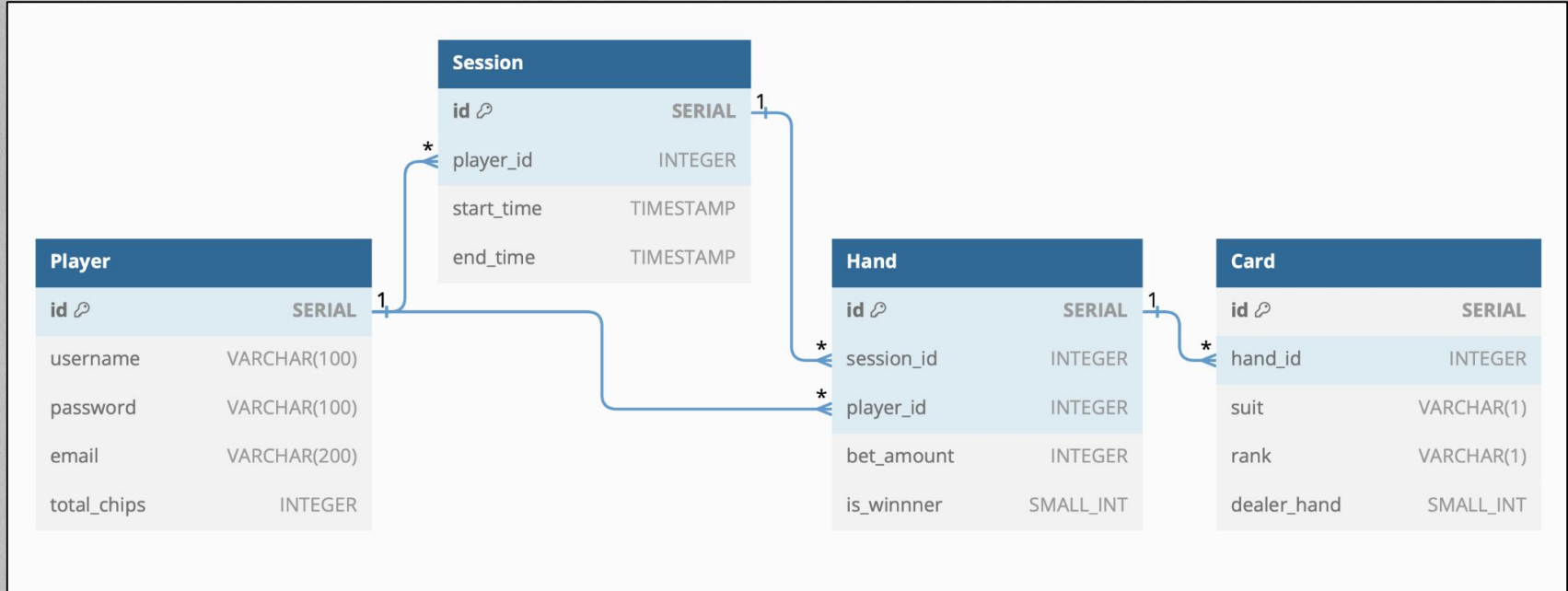




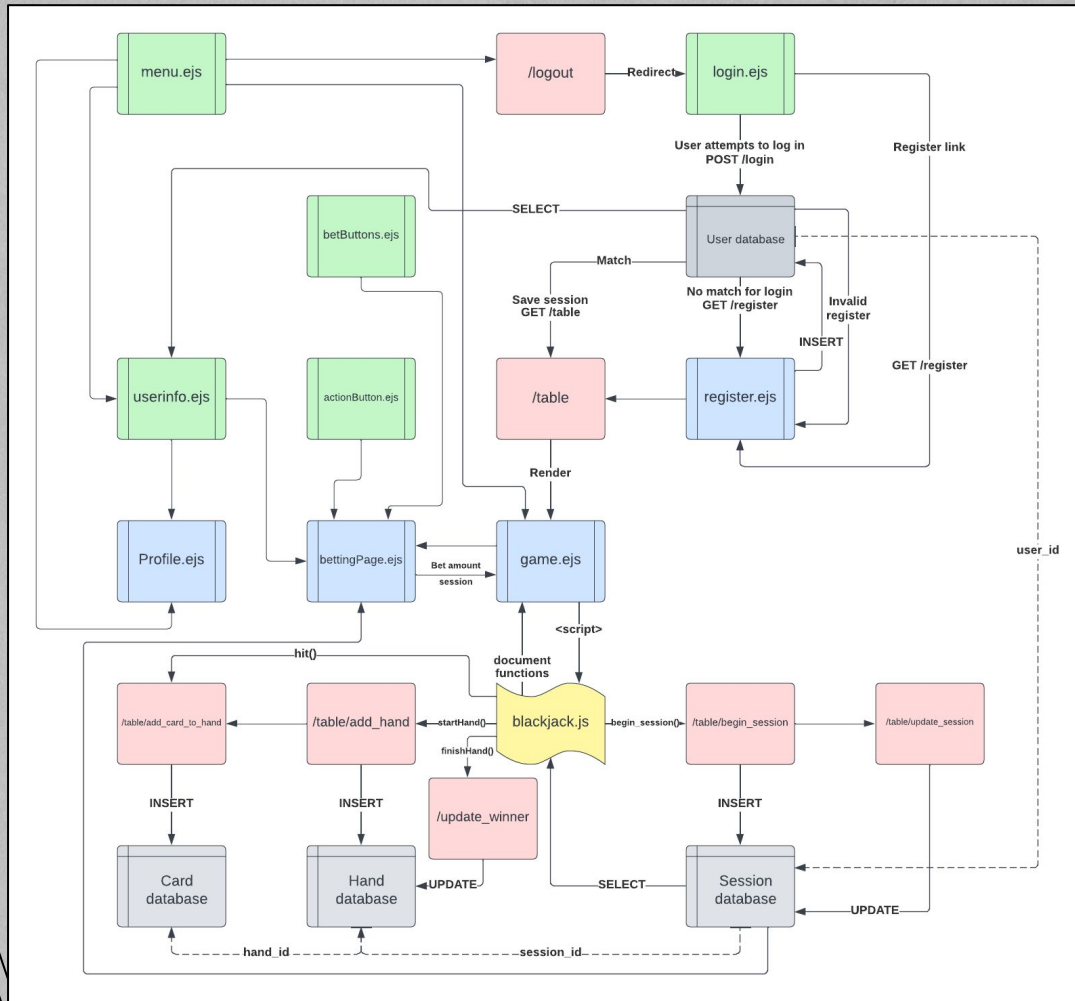
AT Face value (demo)

Blackjack.Net

DaTaBaSe sChEmA



ARCHITECTURE Diagram





IN-DEPTH CODE ANALYSIS

VSCode

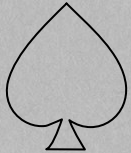
CHALLENGES



01

PROJECT SCOPING

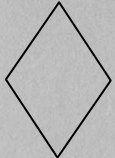
As Rohan said in the beginning, this is crucial



02

COMMUNICATION OF PROGRESS

Communication difficulties lead to repeated work



03

REPOSITORY STANDARDIZATION

Merge conflicts, and functional differences



04

INDIVIDUAL CHALLENGES

Aiden, Alexis, Emanuel, Haoxuan, Zach

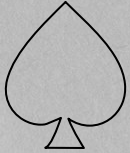
key TakeAways



01

PLAN, PLAN, PLAN

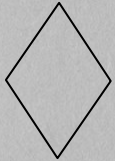
Ensuring a proper scope of the project, before code is ever touched



02

UPLOAD OFTEN & NOTIFY

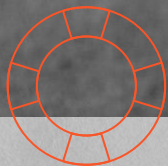
Create effective daily commits that push main forward, and make sure everyone is up to date



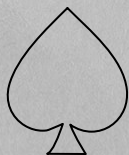
03

STICK TO A STRUCTURE

After planning the architecture, make sure everyone is on the same page



Q&A session



CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon** and infographics & images by **Freepik**