Power4 Web

Power4 Web consists in creating and running a server, in which it will be possible to use a web GUI (graphical user interface).

1. Go Module Initialization

You must initialize your project with a Go module:

```
go mod init power4
```

You will need to create a private repository with the name power4-web

2. Notions

- Golang Documentation: net
- Golang Documentation: ioutil
- Go Web example documentation: templates
- Golang Documentation : templates
- Basic HTML/CSS (no framework)
- Form, get/post

3. Objectives

Build an HTTP server that allows two players to take turns playing Connect Four on the same computer using a local web interface. The game logic will reside entirely in the backend (Go), and the user interface will be generated using Go's HTML templates.

4. Game Mechanics

- The game board is made up of **7 columns and 6 rows**, and is displayed using HTML elements such as <table> or .
- **Each cell** in the grid shows either an empty space or a token representing a player, typically differentiated by **color or shape**.
- To make a move, **players select a column** using a button or an HTML form. This action sends a **POST request** to the server to register the move.
- The **backend manages the game logic**, including alternating turns between Player 1 and Player 2.

- After each move, the page is reloaded to **show the updated board** and indicate which player's turn is next.
- The server checks for two possible outcomes after each move:
    - A win: if **four tokens of the same player are aligned** horizontally, vertically, or diagonally.
    - A draw: if the **grid is completely filled** and no winning alignment is found.

5. HTTP Routes

In this project you will need to implement at least the following endpoints, you can go more if needed:

1. GET /
    - Returns the main HTML page, including the game interface (grid, player info, etc.).
    - *Tip: Use Go templates to dynamically inject data into your HTML page.*
2. POST /play
    - Receives the column number submitted by the player and updates the game state accordingly.
    - *Tip: Use a <form> element in your HTML to allow players to submit their moves. The form's action should point to /play, and the method should be POST.*

6. Instructions

- HTTP server must be written in Go.
- HTML templates must be in project root directory templates.
- The code must respect the good practices.