Challenge: Web Application for Serving Guests

This lesson brings you a challenge to solve.

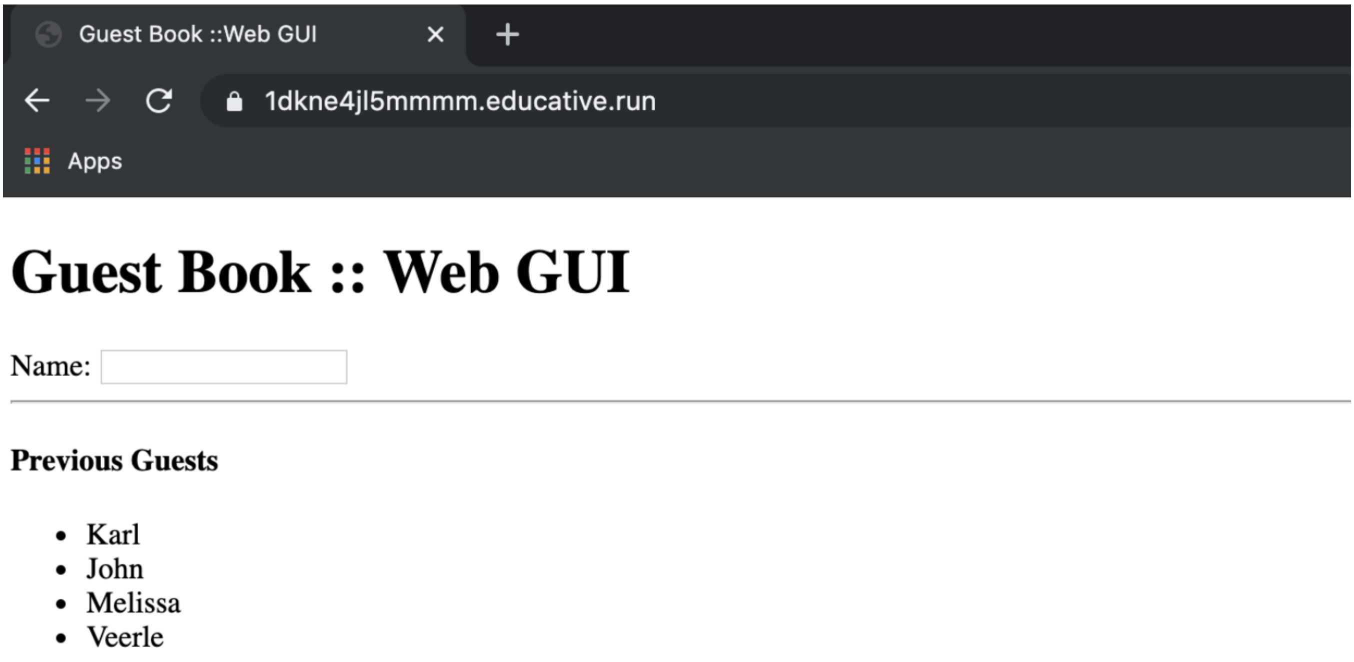
WE'LL COVER THE FOLLOWING

- Problem statement

Problem statement

Implement a simple web application for making a list of *guests* with the following interface:

- An input text field to give the guest-name
- A summed up list of the guest-names already entered
- Also, make it possible to give a name through the url: `/add?name=Vera`



Guest Book :: Web GUI

Name:

Previous Guests

- Karl
- John
- Melissa
- Veerle

Lists of Guests

> **Remark:** Use `0.0.0.0:3000` or simply `:3000` for the connection. However, if you run it locally, use `:8765`

> However, if you run it locally, use `:8080`

Try to attempt the challenge below. Feel free to view the solution, after giving some shots. Good Luck!

> **Hint:** Do not forget to *import* `fmt` and `html/template` packages.

Environment Variables ⌃

Key:	Value:
GOROOT	/usr/local/go
GOPATH	//root/usr/local/go/src
PATH	//root/usr/local/go/src/bin:/usr/local/go...

```
package main
import (
        "net/http"
)

func main() {

}

// indexHandler serves the main page
func indexHandler(w http.ResponseWriter, req *http.Request) {

}

// addHandler add a name to the names list
func addHandler(w http.ResponseWriter, req *http.Request) {

}

var indexHTML = `
<!DOCTYPE html>
<html>
    <head>
                <title>Guest Book ::Web GUI</title>
    </head>
    <body>
                <h1>Guest Book :: Web GUI</h1>
                <form action="/add" method="post">
                Name: <input name="name" /><submit value="Sign Guest Book">
                </form>
                <hr />
                <h4>Previous Guests</h4>
                <ul>
                        {{range .}}
                        <li>{{.}}</li>
                        {{end}}
                </ul>
        </body>
```

```
</html>
`
```

We hope that you were able to solve the challenge. The next lesson brings you the solution to this challenge.