Go In Action 1 Assignment Write-up

1. **Instructions to run the program**
   - This package contains the bookingSystem package, routing package, templates folder and a main.go.
   - Copy the bookingSystem and routing packages to the relevant directory (e.g. $GOROOT/src/Assignment3/).
   - Run the program with "go run main.go" in the terminal or command prompt. Alternatively, open the program in Visual Studio Code and run main.go

2. **Description of the client-server**
   - This application is a dental appointment system which allows users to sign up and log in to the system and search, edit and book dental appointments.
   - Once the server is started up, users can access the application via the client (a web browser) at localhost:5221.
   - The user can sign in with an administrator's credential (admin/password) to access to admin features. Conversely, the user can click on the sign up link to register a new account. A non-admin account will only be able to access user features and not admin features.
   - Logging out will allow the user to switch between admin and non-admin accounts.
   - If an error message shows up on screen, users can go back to the previous page and retry the previous transaction by clicking on the web browser's Back button.

3. **3 features related to concepts discussed in Go In Action 1**
   - HTTP Standard Codes are returned if the server has issues completing a transaction. The ones used here are 303 See Other for redirects, 401 Not Found for failure to obtain results from internal query, 400 Bad Request for incorrect user input and 500 Internal Error for failure to execute templates.
   - The routing package utilises the HandleFunc and ListenAndServer functions from Go's net/http library to match and route different endpoints to functions in the package, as well as to serve the pages via port 5221.
   - Cookies are generated upon login and used for session control. This is used to differentiate between admin and non-admin accounts and serve different features to these 2 groups of users.
   - Data is passed to templates (located in the templates folder) from Go's html/template library to dynamically present information to users e.g. names of patients, appointment details etc.
   - Forms are used to pass data from the client to the server.

4. **3 error handling and concurrency mechanism incorporated**

- Usage of mutex from Go’s sync.Mutex library in Add, Remove, Edit operations (in bookingSystem package) to prevent simultaneous calls to these operations which may lead to a conflict when changing the state of the shared data (e.g. doctor’s availability and appointments information).
- Usage of waitgroups from Go’s sync.WaitGroup (in routing package) to concurrently run goroutine and wait for them to finish.
- Functions defined in the bookingSystem package that fail to obtain results from internal query (e.g. querying for a non-existent appointment or doctor) return error as defined in helper.go in bookingSystem/helperFunctions. This is passed onto the functions in the routing package which uses http.Error() function to wrap these errors.