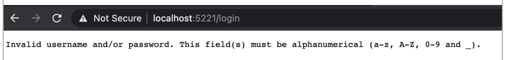
Go In Action 2
Assignment Write-up

1. Instructions to run the program

- This submission contains the packages booking, routing, validation as well as the directories documentation, logs, TLSCerts and templates. It also contains the main.go program and a .env file (which contains the admin user's information).
- Copy the three packages into the relevant directory (e.g. $GOROOT/src/Assignment3/) to run the program.

2. Secure Software Development Techniques applied

<table>
<tr><th>Category</th><th>Description/Discussion</th><th>Relevant Files (if any)</th></tr>
<tr><td><b>Input Validation</b></td><td>Methods in validation package are used to validate if inputs are of the correct format.These include IsDateString, IsTimeString, IsAlphaNumeric and IsAlphabet.
These methods make use of Go's regexp package to match the input patterns.
For example, IsDateString checks if a date is formatted as YYYYMMDD by using the following regular expression.
regexp.MatchString(`^([0-9]{4})(0?[1-9]|1[012])(0?[1-9]|[12][0-9]|3[01])$`, input)</td><td>routing/signup.go
routing/choosePatient.go
routing/searchDoctorAvailability.go
routing/editAppointment.go
routing/searchAvailableDoctors.go
routing/bookNewAppt.go
routing/searchAppointment.go
routing/chooseDoctor.go
routing/login.go</td></tr>
<tr><td><b>Post Validation</b></td><td>If user-input data does not match the expected format, users are informed that submitted data failed to comply with requirements via an error message shown on the client. The same message is also written to the log file.
The following is an example of the error message shown.


Invalid username and/or password. This field(s) must be alphanumerical (a-z, A-Z, 0-9 and _).</td><td>Same as above</td></tr>
</table>

Cross Site Scripting	html/template package is used to sanitise user input to prevent XSS, as it automatically escapes malicious tags such as <script>..</script> and prevents said tags from executing.	All the files in the templates directory.
Communication Security	HTTPS is used to prevent Man-in-the-Middle attacks via Go’s net/http ListenAndServeTLS method. However, because both the public and private keys are generated with openssl and not signed by a Certificate Authority, the application should only be run on localhost (suitable for development/testing only) after configuring the client web browser appropriately.	routing/route.go TLSCerts/cert.pem TLSCerts/key.pem
Error Handling	Package-scope error variables are used in the booking, routing and validation packages. The error messages are shown on the client. The same message is also written to the log files. Access to further resources/navigation forward is denied by default when an error occurs. For errors that are not recoverable, log.Fatal is used to stop the application.	All files in the three packages.
Logging	3 levels of logging (Info, Warning and Error) are used to log event data.	All files in package routing.
Session Management	Cookies are used to maintain a session. A new session is generated upon sign-in and periodic termination of sessions is enforced. Concurrent login is also disallowed. Cookies are also deleted upon logout.	routing/login.go routing/signup.go routing/logout.go
Communicating Authentication Data	Password entry is obscured on screen. Autocomplete for password is also disabled. When handling authentication errors, the application returns "Invalid username and/or password.” so as to obfuscate which is the incorrect field. bcrypt is also used to hash passwords.	templates/signup.gohtml templates/login.gohtml