Go in Action 2

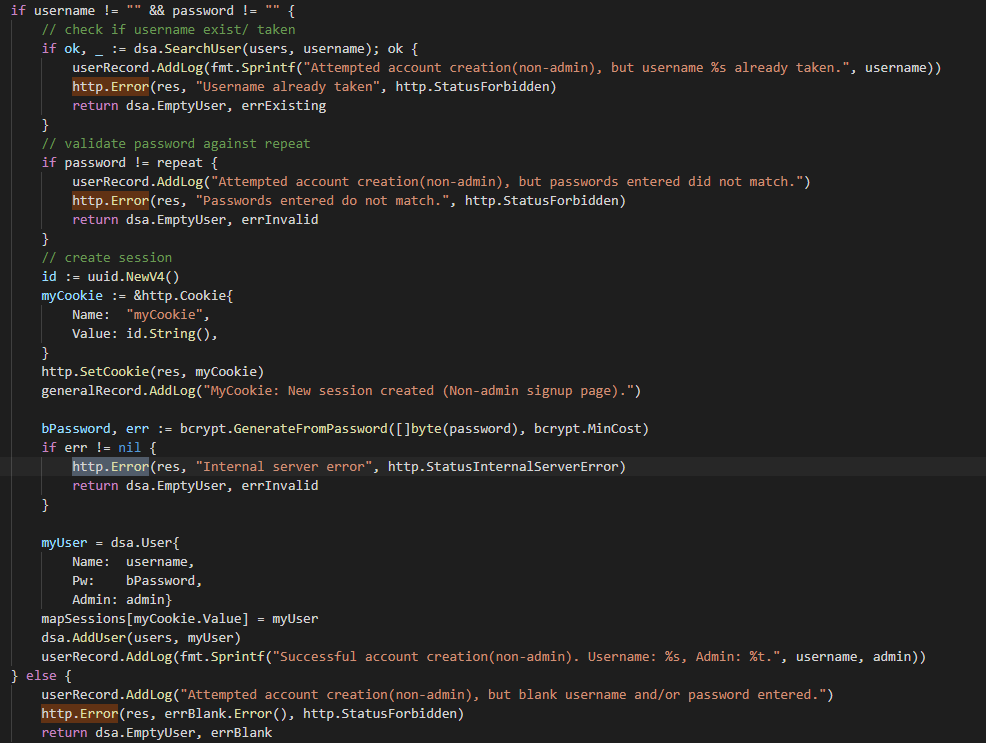
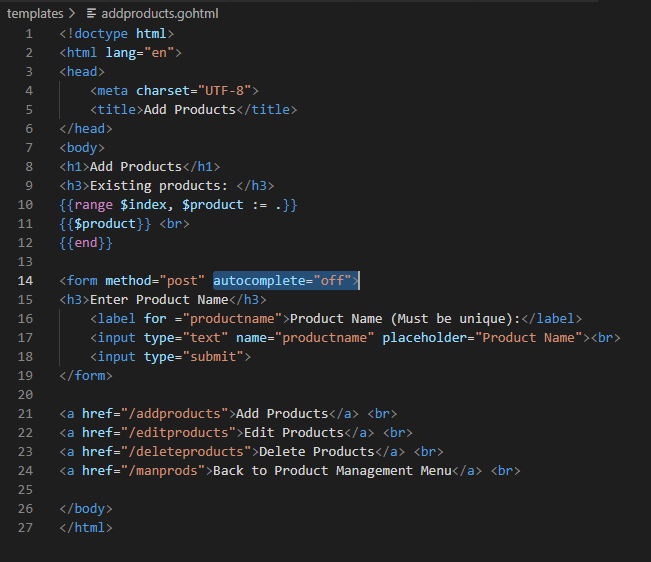
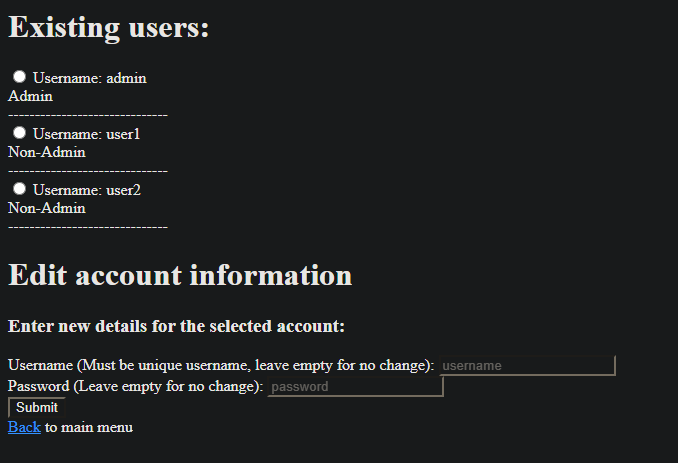
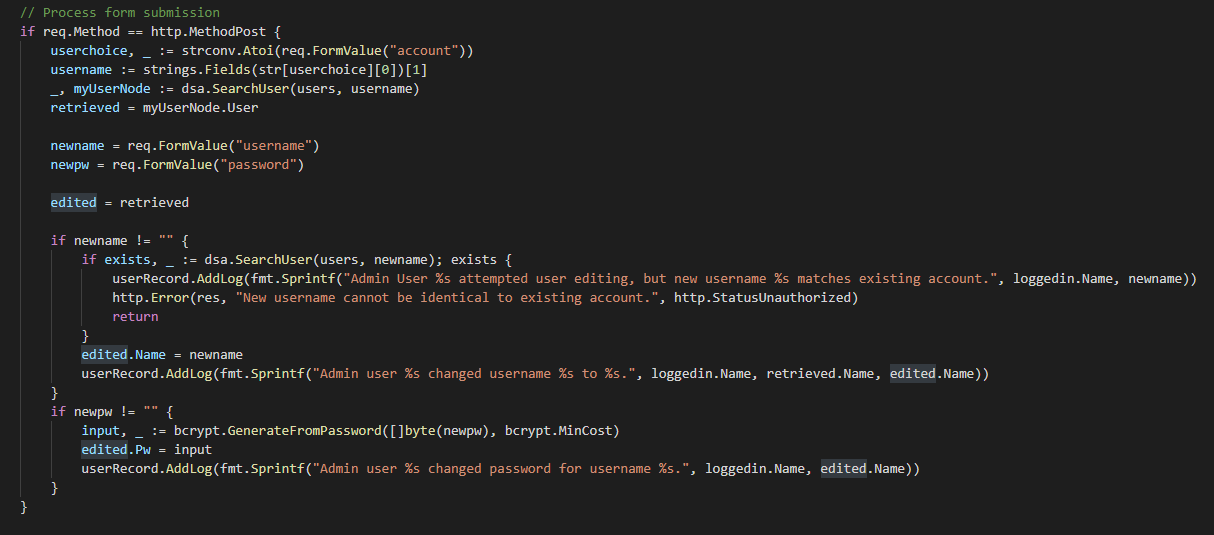
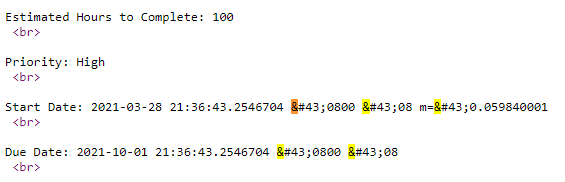
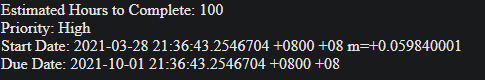
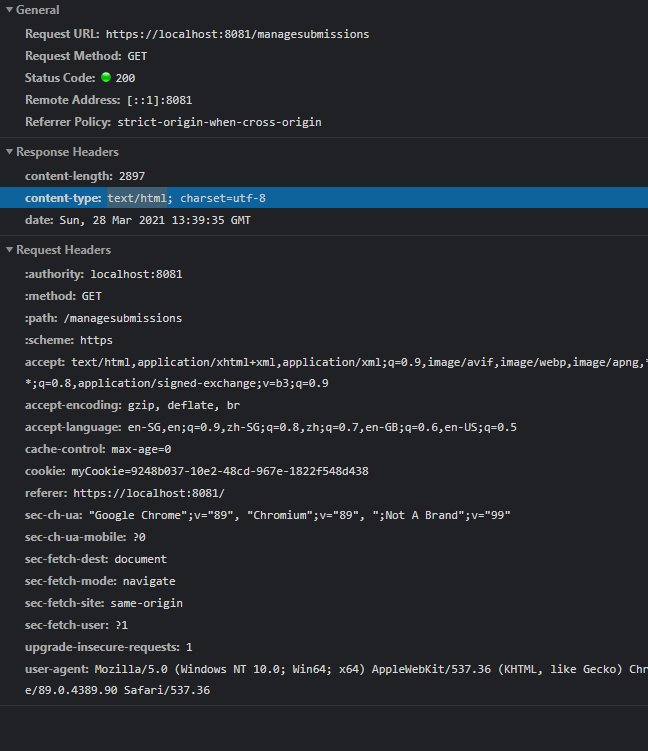
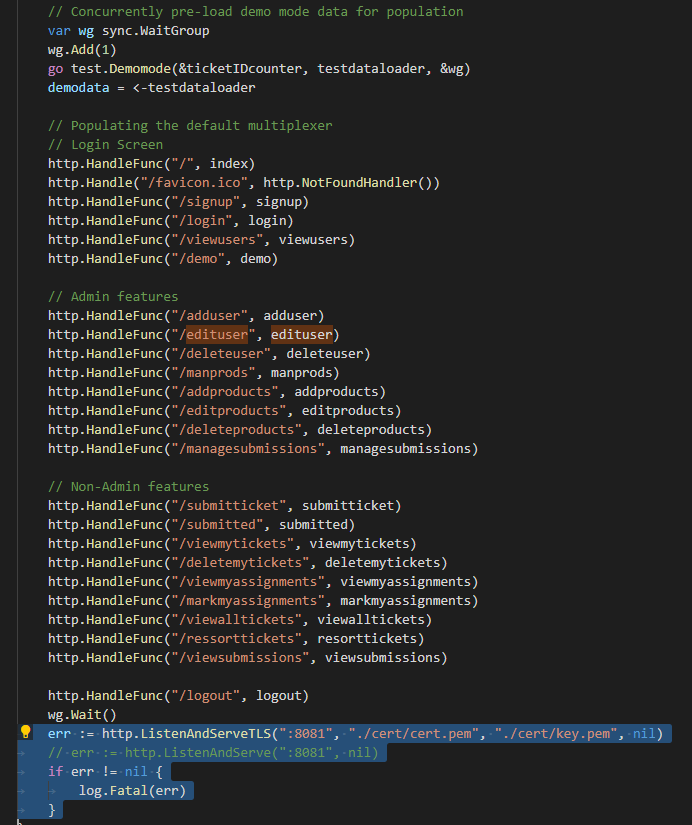
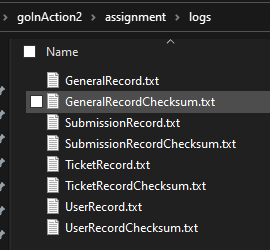
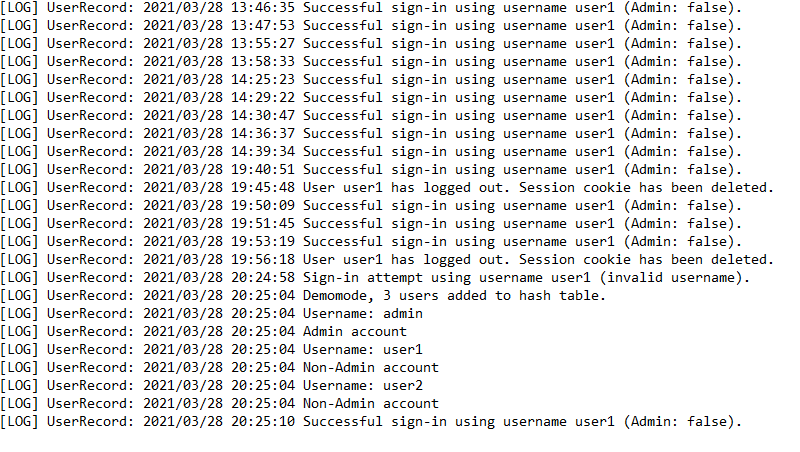
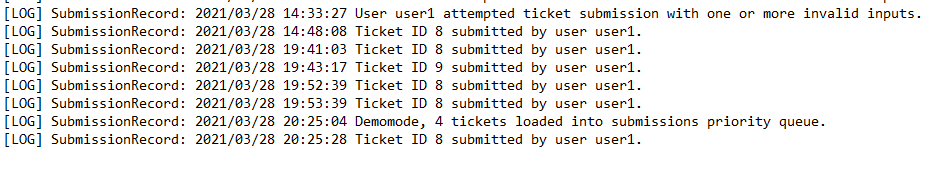
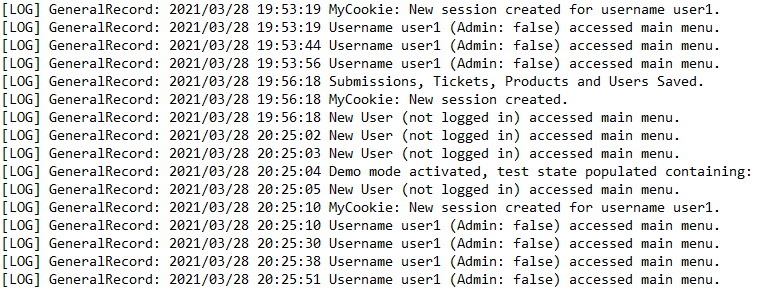
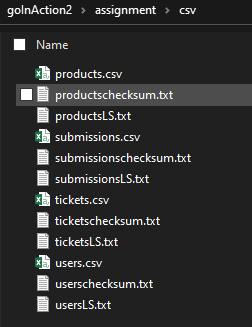
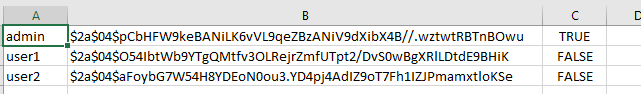
Bug-Tracker Application

Keith Ang

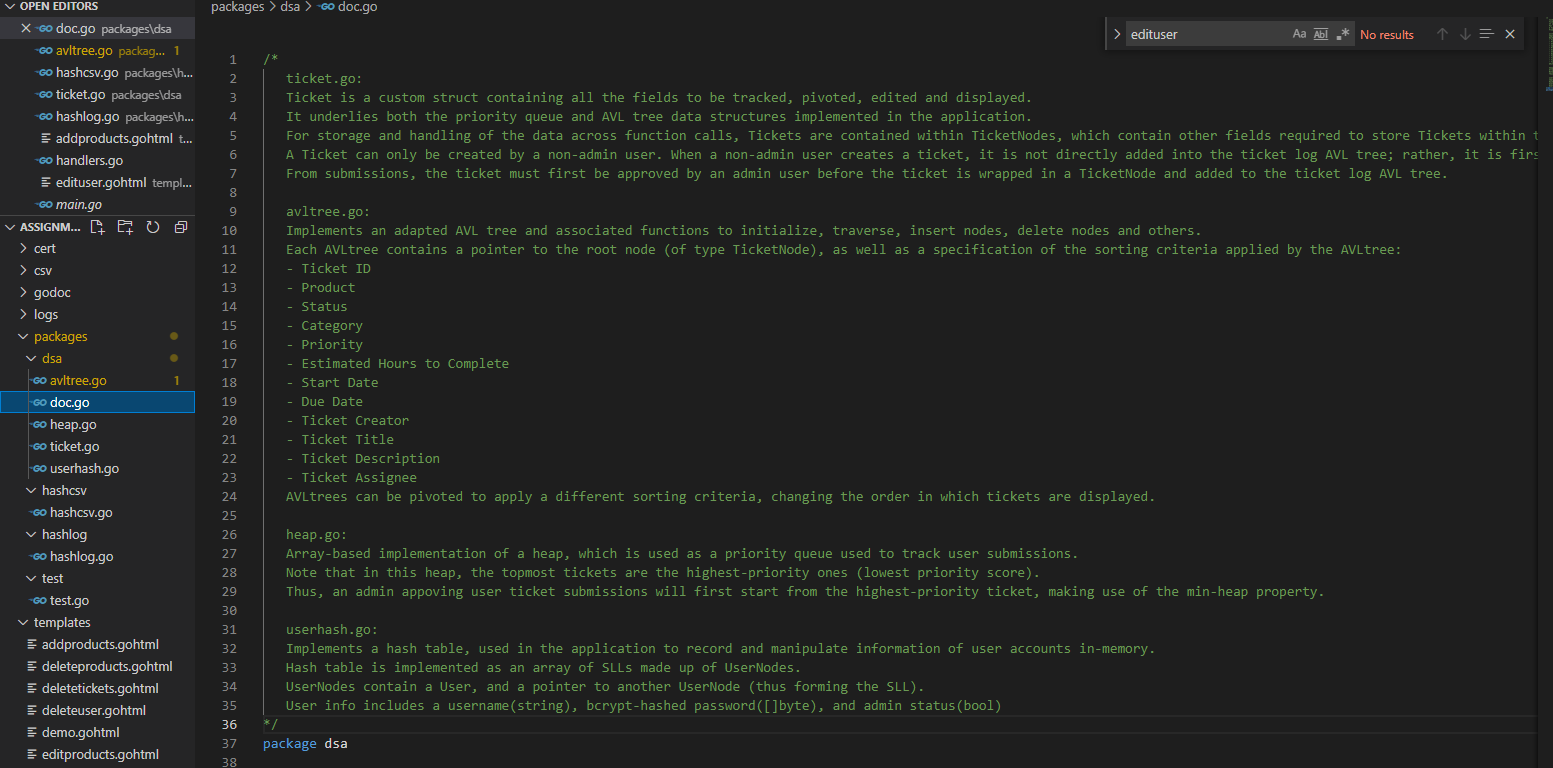
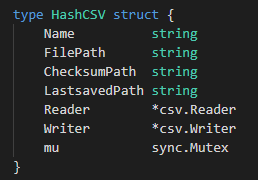
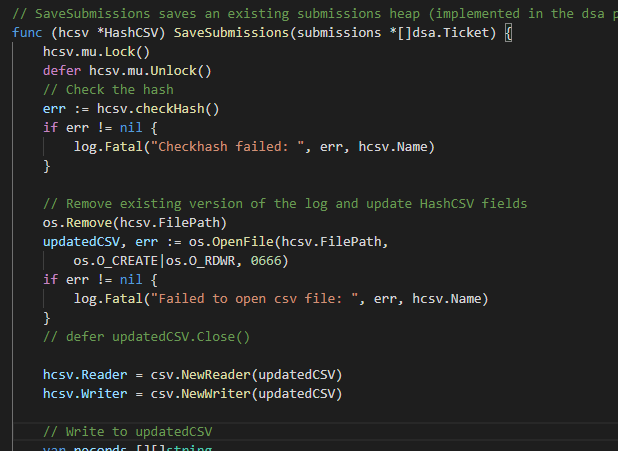
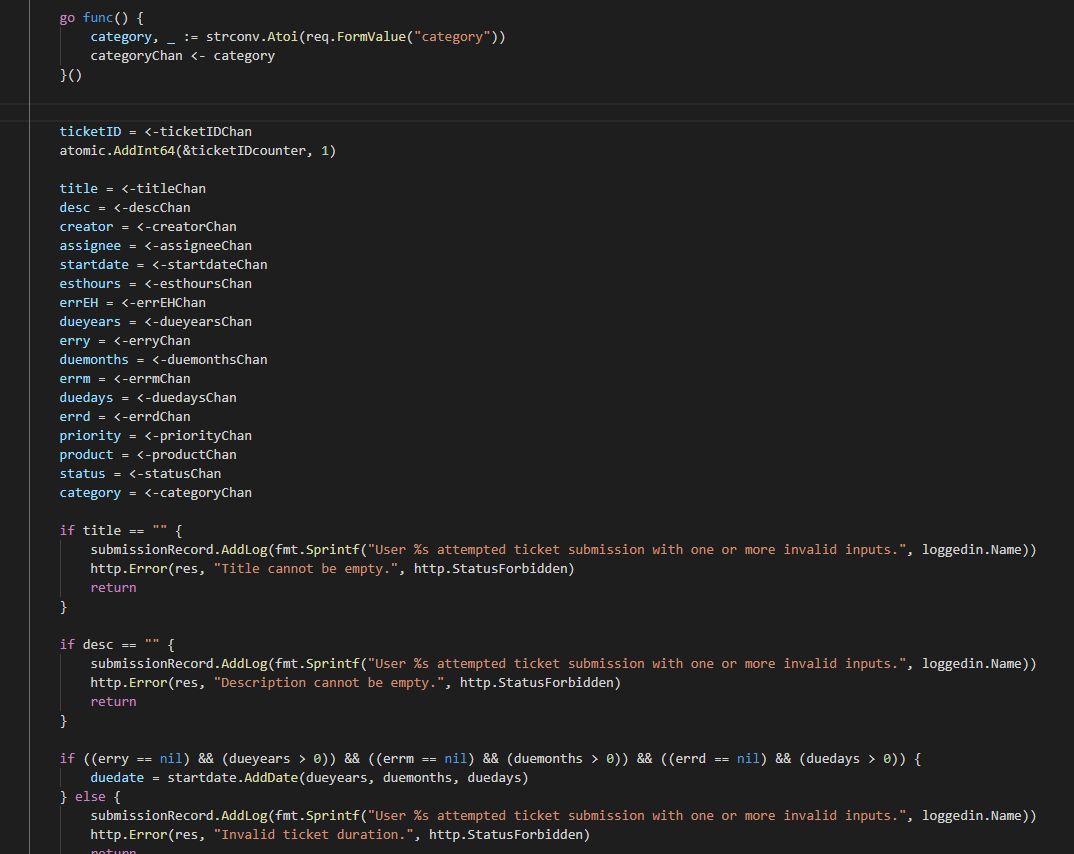
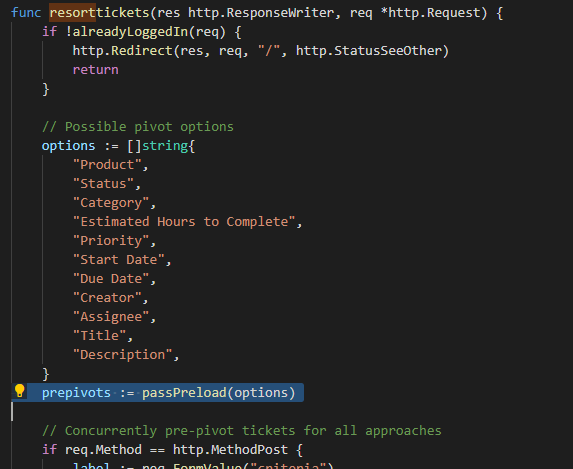
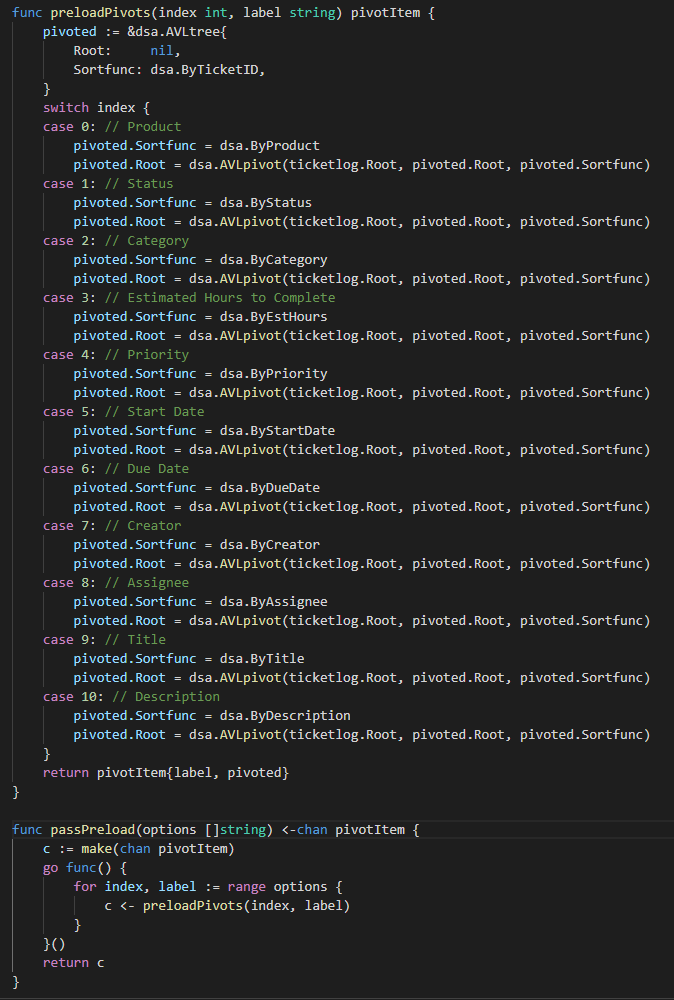
2021

# Secure Software Development Techniques

## Overview

* Input Validation and Sanitization
  + Signing up for new account
  + 
    - Cannot input an existing username
    - Cannot input blank username or password
    - Confirm password by asking user to repeat entry
  + Forms across the application have autocomplete turned off
    - 
  + Data Entry
    - Edit existing account (admin feature):
    - 
    - 
      * Radio select buttons used in forms to facilitate data validation for fields where only one of a view options are valid
      * Validation applied to ensure that username and/or password cannot be blank values
      * Validation applied to ensure that username cannot match an existing user
      * Account username and password can be edited, but admin status cannot be edited (to prevent case of a particular account being granted access to different set of features
* Output Encoding
  + Use of go html/template package to ensure that potential html tags inputted by user are correctly escaped (account creation and editing, product creation and editing, ticket submission)
    - 
    - 
    - 
* Communications Security
  + HTTPS Implementation
    - 
* Error Handling and Logging
  + Logger which records successful (and failed) login attempts
  + 
  + 
  + 
  + 
    - Logger is cryptographically hashed, with SHA256 checksum to verify against
    - Events logged include registering new user accounts (failed and successful)
    - Editing user accounts (failed and successful)
    - Deleting user accounts (failed and successful)
    - Login attempts and successful logins
    - Logouts
    - Ticket submissions and approvals/rejections
    - Cookie creations, and for which user
    - Implementation details found in hashlog source and documentation
* Cryptographic implementations
  + Persistent storage implementation (CSV with SHA256 hash checking)
  + Only hashes of passwords saved
  + Hashed logs and hashed CSV storage are made concurrent safe – see hashcsv and hashlog packages for implementation details
  + 
  + 
* Other security-driven implementations
  + Cookie deleted upon logout

# Idiomatic Go Techniques

* Go Documentation per convention (reference example for multi-file data structures and algorithms package)
  + 
* Mutex hat included in struct
  + 
  + 
* Concurrent design patterns
  + Use of goroutines and channels to concurrently request multiple fields from HTML form
  + 
  + Implementation of generator pattern to concurrently pre-load resorted views of the tickets AVL tree
  + 
  + 

# Documentation of application (Appendix)

Godoc pages for the packages implemented by me for this application, can be found in the ./godoc folder.

Note: In demo mode, user accounts are

* User1 (password: user1)
* User2 (password: user2)
* Admin (password: admin)