| Login testing | | | | |
| --- | --- | --- | --- | --- |
| Code | Input type | input | desired | Fixed how |
| def register():  username = entry\_username.get()  password = entry\_password.get()  if not username or not password:  messagebox.showwarning("Registration Failed", "Username and password cannot be empty.")  return  if os.path.exists('users.txt'):  with open('users.txt', 'r') as file:  users = file.readlines()  for user in users:  stored\_username, \_ = user.split(',')  if stored\_username.strip() == username:  messagebox.showwarning("Registration Failed", "Username already exists!")  return  with open('users.txt', 'a') as file:  file.write(f"{username},{password}\n")  messagebox.showinfo("Registration Success", "Registration successful! You can now log in.") | In bound | qwe,wde | yes | **N/A** |
| On Bound | QAS, WSA | Yes | **N/A** |
| On Bound | 12,34 | Yes | **N/A** |
| Out of Bound | Blank | Yes | **N/A** |
| Adding Assessments | | | | |
| # Dropdown for Assessment number selection  tk.Label(new\_assessment\_frame, text="AS Number:").pack(pady=5)  number\_var = tk.StringVar()  number\_dropdown = ttk.Combobox(new\_assessment\_frame, textvariable=number\_var, values=list(standard\_details.keys()), state="readonly")  number\_dropdown.set("Select AS Number")  number\_dropdown.pack(pady=5) | In Bound | 91521 | Yes | **N/A** |
| 91522 |
| 91523 |
| 91524 |
| 91525 |
| 91526 |
| 91527 |
| # Submit Button  submit\_button = tk.Button(new\_assessment\_frame, text="Submit", command=lambda: add\_assessment(number\_var, name\_var, credits\_var, type\_var, test\_var))  submit\_button.pack(side=tk.BOTTOM, pady=20) | In bound | All fields filled | yes | **N/A** |
| Out of Bound | Blank | Yes | **N/A** |
|
| Adding Assessment to home page | | | | |
| for assessment in assessments\_list:  tree.insert("", "end", values=(  assessment["AS Number"],  assessment["Name"],  assessment["Credits"],  assessment["Type"],  assessment["Due Date"],  assessment["Test Type"]  )) | On Bound | 91521 | Yes | **N/A** |
| 91522 |
| 91523 |
| 91524 |
| 91525 |
| 91526 |
| 91527 |
| Out of Bound | Pressed submit twice | No, would add the same assessment and data twice. | Needed to edit the function to check whether the as number was already there  \* also added ability to make changes |
| Out of Bound | Pressed submit without filling anything out | No, need it to come back with an error but it didn’t | Needed to add an if statement to check that all the fields are filled. |

| Problem code | Problem | Solution |
| --- | --- | --- |
| number\_dropdown = ttk.Combobox(root, textvariable=number\_var, values=[AS\_NUMBERS], state="readonly") | Would display all standards in a line | Had to change values to the individual numbers rather than the dictionary AS\_NUMBERS |
| tk.Label(new\_assessment\_frame, text="AS Name:").pack(pady=5)  name\_var = tk.StringVar()  name\_dropdown = ttk.Combobox(new\_assessment\_frame, textvariable=name\_var, values=[details["Name"] for details in standard\_details.values()], state="readonly")  name\_dropdown.set("Select AS Name")  name\_dropdown.pack(pady=5) | Would cut off the names of the assessments. | Had to change the max length to set dynamically. |
| max\_length = max(len(name) for name in standard\_details.values())  tk.Label(new\_assessment\_frame, text="AS Name:").pack(pady=5)  name\_var = tk.StringVar()  name\_dropdown = ttk.Combobox(new\_assessment\_frame, textvariable=name\_var, values=[details["Name"] for details in standard\_details.values()], state="readonly", width=max\_length)  name\_dropdown.set("Select AS Name")  name\_dropdown.pack(pady=5) | Made the box even smaller | Had to change  **max\_length = max(len(name) for name in standard\_details.values())**  to  **max\_length = max(len(details["Name"]) for details in standard\_details.values())** |
| def page\_menu():  def collapse\_menu():  toggle\_menu.destroy()  toggle\_button.configure(text='☰')  toggle\_menu = tk.Frame(root, bg='red')  window\_height = root.winfo\_height()  window\_width = root.winfo\_width()  toggle\_menu.place(x=0, y=50, height=window\_height, width= window\_width \* 0.25)  toggle\_button.config(text ="X" )  toggle\_button.config(command=collapse\_menu) | Would open the assessments page on the same page multiple times | Had to add a function to destroy the widget if it was already there. |
| if selected\_number == 91521:  name\_var.set(standard\_details"Name")  credits\_var.set(standard\_details"Credits")  type\_var.set(standard\_details"Type") | Would come up with error-  name\_var.set(standard\_details"Name")  ^^^^^^  SyntaxError: invalid syntax | Added [] around “Name” etc |
| if selected\_number == 91521:  name\_var.set(standard\_details["Name"])  credits\_var.set(standard\_details["Credits"])  type\_var.set(standard\_details["Type"]) | Error -  name\_var.set(standard\_details["Name"])  ~~~~~~~~~~~~~~~~^^^^^^^^  KeyError: 'Name' | Fixed by changing **if selected\_number ==**... to  **if selected\_standard in standard\_details:** |
| def side\_bar():  head\_frame = tk.Frame(root, bg='red')  head\_frame.pack(side=tk.TOP, fill=tk.X)  head\_frame.configure(height=50) | Side bar would show up at the bottom | Had to call the sidebars function before the login frame |
| toggle\_menu = tk.Frame(root, bg='red')  toggle\_menu.place(x=0, y=0, height=500, width= 500) | Would overlap the header | Had to change **y=0** to **y=50** |
| def side\_bar():    # designing the frame  head\_frame = tk.Frame(root, bg='red', highlightbackground='white', highlightthickness=10)  toggle\_button = tk.Button(head\_frame, text='☰', fg="white", bg="red", font=20)  toggle\_button.pack(side=tk.LEFT)  head\_frame.pack(side=tk.TOP, fill=tk.X)  head\_frame.pack\_propagate(False)  head\_frame.configure(height=50)  toggle\_menu = tk.Frame(root, bg='red')  toggle\_menu.place(x=0, y=50, height=500, width= 500) | The menu bar would appear continuously and there was no way to get rid of it | Had to make  **toggle\_menu = tk.Frame(root, bg='red')**  **toggle\_menu.place(x=0, y=50, height=500, width= 500)**  Into a function |
| ‘’ | Wouldn’t reach the whole page | Had to change height to a variable and use **root.winfo\_height()** to get the height of the screen |
| def page\_menu():  def collapse\_menu():  toggle\_menu.destroy()  toggle\_button.configure(text='☰')  toggle\_menu = tk.Frame(root, bg='red')  window\_height = root.winfo\_height()  window\_width = root.winfo\_width()  toggle\_menu.place(x=0, y=50, height=window\_height, width= window\_width \* 0.25)  toggle\_button.config(text ="X" )  toggle\_button.config(command=collapse\_menu) | **Toggle\_button** would have a yellow underline meaning that it was undefined and code wouldn’t work | Had to make it a global function  **global toggle\_button** |
| ‘’ | Side bar would only open once | Had to create separate functions for the opening and closing of the menu so that it can be called multiple times. |
| def page\_menu():  def collapse\_menu():  global toggle\_menu  toggle\_menu.pack\_forget() toggle\_button.config(text='☰') toggle\_button.config(command=expand\_menu)  def expand\_menu():  global toggle\_menu  if toggle\_menu is None:  toggle\_menu = tk.Frame(root, bg='#253B80')  toggle\_menu = tk.Frame(root, bg='#253B80')  main\_btn = tk.Button(toggle\_menu, text='Home', font=('Bold', 20), bg='#253B80', fg='black', command=show\_assessment\_page, relief="flat")  main\_btn.place(x=20, y=20)    as\_btn = tk.Button(toggle\_menu, text='New Assessment', font=('Bold', 20), bg='#253B80', fg='black', command=new\_assessment, relief="flat")  as\_btn.place(x=20, y=140)  window\_height = root.winfo\_height() window\_width = root.winfo\_width() toggle\_menu.place(x=0, y=50, height=window\_height, width=window\_width \* 0.25)  toggle\_button.config(text="X")  toggle\_button.config(command=collapse\_menu) expand\_menu()  toggle\_button.config(command=collapse\_menu) | Now it wouldn’t close at all | Had to change **toggle\_menu.pack\_forget()**  to  **toggle\_menu.place\_forget()**  So that it would forget where it was placed rather than the whole function |
| # Create buttons for pages  as\_btn = tk.Button(toggle\_menu, text='New Assessment', font=('Bold, 20'), fg='white', command=new\_assessment)  as\_btn.place(x=20,y=140) | Would move the menu bar to the bottom of the page | Needed to move the call function of the sidebar so that it would get added before the widgets. |
| submit\_button = tk.Button(new\_assessment\_frame, text="Submit", command=lambda: add\_assessment(number\_var, name\_var, credits\_var, type\_var))  submit\_button.pack(side=tk.BOTTOM, pady=20) | Would print the test type as external rather than test or essay | Had to add **test\_var** to the def new assessments function to solve |
| entry\_username = tk.Entry(root, bg="#B02324", text= "\_\_\_\_\_\_\_\_\_\_\_", relief="flat") | Wouldn’t have a line underneath and the whole box would just disappear | Had to add a separate frame to act as the line. |
| entry\_username = tk.Entry(root, bg="#B02324", relief="flat")  entry\_username.pack(pady=10)  username\_underline = tk.Frame(root, height=2, bg="#E6E6E6")  username\_underline.pack(fill="x", padx=20, pady=(0, 10)) | There would be a space between the line and the actual box | Had to get rid of the padding for the entry box section |
| entry\_username = tk.Entry(root, bg="#B02324", relief="flat")  entry\_username.pack()  username\_underline = tk.Frame(root, height=2, bg="#E6E6E6")  username\_underline.pack(fill="x", padx=20, pady=(0, 10)) | The input wouldn’t go to the end of the line and was centered to the middle | Had to match the **fill** and **padx** so that the line started and ended where the input box was |
| if not assessments\_list:  return  upcoming = []  today = datetime.today()  for assessment in assessments\_list:  try:  due\_date = datetime.strptime(assessment["Due Date"], "%d-%m-%Y")  days\_left = (due\_date - today).days  if days\_left == 1:  upcoming.append(f"{assessment['Name']} is due in {days\_left} day(s) on {assessment['Due Date']}")  except Exception as e:  print("Error parsing date:", assessment["Due Date"], e)  if upcoming:  messagebox.showwarning("Upcoming Assessments", "\n".join(upcoming)) | No pop up would show | There was no need to append, once i got rid of it, it worked fine |
| def expand\_menu():  global toggle\_menu  # Check if sidebar is already created (so it doesn't recreate it)  if toggle\_menu is None:  toggle\_menu = tk.Frame(root, bg='#253B80')  # Create actual menu that will appear when pressed  toggle\_menu = tk.Frame(root, bg='#253B80')  # Create buttons once  main\_btn = tk.Button(toggle\_menu, text='Home', font=('Bold', 20), bg='#253B80', fg='white', command=show\_assessment\_page, relief="flat")  main\_btn.place(x=20, y=20)  as\_btn = tk.Button(toggle\_menu, text='New Assessment', font=('Bold', 20), bg='#253B80', fg='white', command=new\_assessment, relief="flat")  as\_btn.place(x=20, y=140)  bottom\_bar = tk.Frame(toggle\_menu, bg='orange', height=80)  bottom\_bar.pack(side=tk.BOTTOM, fill=tk.X)  logout\_btn = tk.Button(bottom\_bar, text='Logout', font=('Bold', 20), bg='#253B80', fg='white', command=logout, relief="flat")  logout\_btn.place(x=20, y=10)  window\_height = root.winfo\_height()  window\_width = root.winfo\_width()  toggle\_menu.place(x=0, y=50, height=window\_height, width=int(window\_width \* 0.25))  toggle\_button.config(text="X", command=collapse\_menu) | The Side menu wouldn’t work at all, wouldn’t show up when pressed | Needed to get rid of the **toggle\_menu = tk.Frame(root, bg='#253B80')**  Because it showed up twice so it wasn’t forgetting the frame once it was made, just forgot it completely. |
| def expand\_menu():  global toggle\_menu  if toggle\_menu is None:  toggle\_menu = tk.Frame(root, bg='#253B80')  # Create buttons once  main\_btn = tk.Button(toggle\_menu, text='Home', font=('Bold', 20), bg='#253B80', fg='white', command=show\_assessment\_page, relief="flat")  main\_btn.place(x=20, y=20)  as\_btn = tk.Button(toggle\_menu, text='New Assessment', font=('Bold', 20), bg='#253B80', fg='white', command=new\_assessment, relief="flat")  as\_btn.place(x=20, y=140)  bottom\_bar = tk.Frame(toggle\_menu, bg='orange', height=80)  bottom\_bar.pack(side=tk.BOTTOM, fill=tk.X)  logout\_btn = tk.Button(bottom\_bar, text='Logout', font=('Bold', 20), bg='#253B80', fg='white', command=logout, relief="flat")  logout\_btn.place(x=20, y=10)  window\_height = root.winfo\_height()  window\_width = root.winfo\_width()  toggle\_menu.place(x=0, y=50, height=window\_height, width=int(window\_width \* 0.25))  toggle\_button.config(text="X", command=collapse\_menu) | The logout button was too low and cut off majority of the button | Had to change **toggle\_menu.place(x=0, y=50, height=window\_height, width=int(window\_width \* 0.25)** as it was pushing the logout buttons frame down in order to fit the rest of it in. |