UI/UX Design with Streamlit

Start Assignment

Due Apr 23 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip
Available Mar 16 at 12am - Apr 23 at 11:59pm

Introduction and Motivation

A web application is an interactive computer program developed using web technologies (HTML, CSS, JS), which stores (Database, Files) and manipulates data (CRUD - create, read, update and delete). Webapps can be used by a team or single user to perform tasks over the internet.

Objective: In this assignment, you will develop a web application using the Streamlit framework that focuses on usability goals and adheres to human-computer interaction (HCI) design principles. You will use Streamlit features and API requests to fetch and display data in various forms, such as charts, plots, maps, and tables. Your web app must include interactive widgets, such as buttons, select boxes, radio buttons, text input, color input, and other features that help achieve the HCI guidelines. Your project needs to manipulate data requested through an API. The format of the documents received can be of any type, but I highly recommend JSON, XML, or CSV.

Assignment Details

1. **Select a theme or topic for your web application:** Choose a topic that interests you and has public APIs available for data retrieval. Examples include weather forecasting, stock market analysis, social media trends, or public transportation information.
2. **Research and identify relevant APIs**: Find one or more APIs that provide data relevant to your chosen topic. Ensure the APIs are publicly accessible and have clear documentation.

   Here is a list of possible APIs that might be of interest to you:

   - https://github.com/public-apis/public-apis
   - https://rapidapi.com/collection/list-of-free-apis
3. **Define usability goals:** Based on your selected topic, define a set of usability goals for your web app. These may include effectiveness, efficiency, learnability, memorability, error prevention, and user satisfaction.
4. **Design the web application:** Sketch out a rough layout of your application, identifying the key components and user interactions. Consider the placement of widgets, navigation elements, and data visualizations.
5. **Develop the web app using Streamlit**: Implement your design using Streamlit, incorporating the following features:
   1. API requests: Fetch data from the selected APIs
   2. At least 1 **interactive table** (https://docs.streamlit.io/library/api-reference/data/st.dataframe)
   3. At least 2 **chart elements**, such as line, area or bar charts (matplotlib is allowed). To display:
      1. a **line chart** - https://docs.streamlit.io/library/api-reference/charts/st.line_chart
      2. an **area chart** - https://docs.streamlit.io/library/api-reference/charts/st.area_chart
      3. a **bar chart** - https://docs.streamlit.io/library/api-reference/charts/st.bar_chart
   4. At least 1 **map** with points marked on it (a simple map can be created using https://docs.streamlit.io/library/api-reference/charts/st.map or a more complex 3d map at https://docs.streamlit.io/library/api-reference/charts/st.pydeck_chart)
   5. At least 1 **button widget** (https://docs.streamlit.io/library/api-reference/widgets/st.button)
   6. At least 1 **checkbox widget** (https://docs.streamlit.io/library/api-reference/widgets/st.checkbox)
   7. At least 2 of the essential feedback and messages boxes to the users:
      1. **Success box** - https://docs.streamlit.io/library/api-reference/status/st.success
      2. **Information box** - https://docs.streamlit.io/library/api-reference/status/st.info
      3. **Warning box** - https://docs.streamlit.io/library/api-reference/status/st.warning
      4. **Error box** - https://docs.streamlit.io/library/api-reference/status/st.error
      5. **Exception message (optional)** - https://docs.streamlit.io/library/api-reference/status/st.exception
   8. At least any 5 different widgets chosen from the following:
      1. **Radio button** - https://docs.streamlit.io/library/api-reference/widgets/st.radio
      2. **Selectbox** - https://docs.streamlit.io/library/api-reference/widgets/st.selectbox
      3. **Multiselect** - https://docs.streamlit.io/library/api-reference/widgets/st.multiselect
      4. **Slider** - https://docs.streamlit.io/library/api-reference/widgets/st.slider
      5. **Select-slider** - https://docs.streamlit.io/library/api-reference/widgets/st.select_slider
      6. **Text input** - https://docs.streamlit.io/library/api-reference/widgets/st.text_input
      7. **Number input** - https://docs.streamlit.io/library/api-reference/widgets/st.number_input
      8. **Text-area** - https://docs.streamlit.io/library/api-reference/widgets/st.text_area
      9. **Date input** - https://docs.streamlit.io/library/api-reference/widgets/st.date_input
      10. **Time input** - https://docs.streamlit.io/library/api-reference/widgets/st.time_input
      11. **File uploader** - https://docs.streamlit.io/library/api-reference/widgets/st.file_uploader
      12. **Color** - https://docs.streamlit.io/library/api-reference/widgets/st.color_picker
   9. You may include a progress bar for certain components of your application; however, this is not mandatory
   10. You may include media elements such as image, audio or video, which are not a requirement but can add to the overall harmony of the web application being developed
   11. Streamlit allows you to display a sidebar, insert containers laid out as side-by-side columns, insert a multi-element container that can be expanded/collapsed, among many other features:

       Sidebar - https://docs.streamlit.io/library/api-reference/layout/st.sidebar
       Columns - https://docs.streamlit.io/library/api-reference/layout/st.columns
       Expander - https://docs.streamlit.io/library/api-reference/layout/st.expander
6. **Apply HCI design principles:** Ensure your web app adheres to HCI design principles, such as visibility, feedback, consistency, flexibility, and error prevention. Consider the following aspects:
   a. Information architecture: Organize content and functionality to promote ease of use and understanding.
   b. Navigation: Provide clear and consistent navigation options.
   c. User feedback: Offer immediate and informative feedback to user actions.
   d. Aesthetics: Maintain a visually appealing and professional design.
7. **Test your web app**: Conduct usability testing with a small group of users to identify any issues or areas for improvement. Make any necessary changes based on the feedback received.
8. **Document your work:** Prepare a report that includes the following sections:
   a. Introduction: Introduce your web app and its purpose.
   b. Usability goals: Describe the usability goals you set for your web app and explain how you addressed each goal.
   c. Design process: Discuss your design process, from sketching to implementation.
   d. API integration: Explain how you utilized the APIs and discuss any challenges or limitations encountered.
   e. Interactive widgets: Describe the widgets you incorporated and their purposes.
   f. HCI design principles: Discuss how your web app adheres to HCI design principles.
   g. Testing and feedback: Summarize the results of your usability testing and any changes made in response to feedback.
   h. Conclusion: Reflect on your experience and discuss potential future improvements.
9. **Submission:** Submit your web app's source code, along with your report and any supporting materials, such as sketches or user feedback.

Requirements

The web application needs to be developed using Python programming languages version 3.9 (or later) and the open-source framework Streamlit. Your objective is to create a web application that manipulates **data** and displays **information** to the users.

The biggest HCI challenges will be the selection of appropriate layouts and containers. For example, you will need to study your target audience and the data being manipulated to decide, for example, whether to have 3 columns or 1 single column, or even whether an input data should be entered as a radio button or a text field, or whether to display a certain data output as a line chart or a map. Ask yourselves throughout the development process some of these questions:

Initial Steps

1. Choose a topic of interest to you (*examples: sports, geology, music, cryptocurrency, farming, etc.*)
2. Search for free and public datasets or free public APIs where you can request data to be manipulated and visualized in your web application
3. Start developing your web application:
   1. Download and install Python programming language version 3
   2. Download and Install PyCharm IDE Professional version by applying for a student license using your FIU email
   3. Create a new Python project in PyCharm IDE and:
      1. Install the necessary packages: **pip install streamlit numpy pandas**
      2. Import the above packages:
         import streamlit as st
         import pandas as pd
         import numpy as np
1. Include the components listed in REQUIREMENTS according to the goal of your web application and to the type of data being manipulated.
2. Distribute text elements in the form of title, header, subheader, caption or pre-formatted text across the web application so that the content of your web app engages readers and drives them to taken the necessary actions to fulfill your app's goals. Avoid dull, lifeless, and overlong prose. Try keeping text short and intriguing. This will encourage users to click through to other elements. Group content into cohesive categories by breaking it up into short paragraphs enriched with visual elements. This can help you make your web app have a light and engaging feel.
3. You are ready to submit it!
4. Don't worry. I know this was so much fun. But this is not the end of this project. You will need it for our final and, arguably the most important branch of HCI, project, the Usability Test Project.

Grading Criteria

Your project will be graded based on the following criteria:

1. Relevance and clarity of the chosen topic.
2. Successful integration of APIs and data visualization.
3. Inclusion and effective use of interactive widgets.
4. Adherence to HCI design principles.
5. Quality of testing and responsiveness to user feedback.
6. Thoroughness

This is how you will receive a score for your project submission:

Criteria	Exemplary (10)	Proficient (8)	Adequate (6)	Insufficient (4)
Identification of target users	Users clearly identified and web app clearly caters for them	Users partially identified, but web app caters for this group	Users identified but web application does not necessarily cater for them	No users identified or mentioned in the project submission
Goal Identification	The web app's purpose is readily apparent to the user.	The web app has a firm purpose, but may occasionally digress from the purpose	The purpose is not always clear	The purpose is generally unclear
Link between the goal Identification and the target audience	The problem poses a novel perspective or a major opportunity for innovation	Web app provides firm support for the goal identification and displays evidence of a basic analysis of a sufficiently limited topic. User gains some insights	The problem is somewhat clear but solution feels like a hammer trying to find a nail	The problem is unclear, unimportant or not well-motivated.
Dataset properly identified	Dataset clearly identified and used either through pre-selected XML, CSV or JSON files or through API requests	Dataset is identified and used either through pre-selected XML, CSV or JSON files or through API requests, but insufficient for the goal of the application	Dataset not properly used for the web app's goal	No dataset identified
Dataset properly manipulated	Data was correctly manipulated from the requested documents using the API	Data was correctly manipulated from the requested documents using the API; however, it did not match with the goal identified for this web app	The data was not properly managed from the JSON, CSV, or XML files.	No dataset identified
Execution	The web app is ready to be deployed and used	All relevant aspects of the project have been completed, but they have minor flaws	The basic elements of the research are complete, but either they are flawed or important aspects are still missing	The project has some complete components, but critical aspects are incomplete
Organization	The ideas are arranged logically to support the web app's goal. They flow smoothly from one to another and are clearly linked to each other. User can follow line of reasoning.	The ideas are arranged logically to support the web app's goal. They are usually clearly linked to each other. For the most part, user can follow line of reasoning.	The elements of the web app are not arranged logically. Frequently, ideas fail to make sense together. User can figure out what the developer probably intends but may not be motivated to do so.	The elements of the web application lacks any semblance of logical organization. The user cannot identify a line of reasoning and loses interest.
Streamlit properly used	All elements of Streamlit listed in the requirements were used	Not all elements of Streamlit listed in the requirements were used	Just some elements of Streamlit listed in the requirements were used	Just a few elements of Streamlit listed in the requirements were used
Submission	On time	1 day late	2 days late	4 days late
Presentation	Web application was presented well with HCI elements clearly identified and correctly used	Web application was presented well, but HCI elements were not clearly identified or correctly used	Presentation lacked organization, identification of HCI elements, and did not meet the requirements	No presentation of the web application