

## Homework 6: Server-Side Scripting using Python, JSON and eBay API

### Grading Guidelines

#### HTML, CSS and JavaScript Functionality:

1. Search Form (3.25 points in total)
  - The initial webpage should load with proper fields and defaults in the search form, along with the “eBay shopping” header - **1** point
  - Proper styling of the form, text alignment of fields, according to video - **0.25** points
  - Search should support checking multiple conditions together – **0.25** points
  - The eBay header image should redirect to the eBay shopping website on click – **0.25** points
  - **JavaScript validation** for the search input- Keyword, Min Price and Max Price (**1** point)
    - All the validations (Empty keyword,  $\text{MaxPrice} \geq \text{MinPrice}$ ,  $\text{MinPrice} > 0$ ,  $\text{MaxPrice} > 0$ ) should function correctly - **0.75** points
    - The page maintains the previous input values - **0.25** points
  - “**Clear**” button clears all input textboxes, checkboxes, and sets correct default values for “Sort By”. (**0.5** point)
    - If your page fails to clear one of the fields, 0.25 point will be deducted.
2. Card Design (2.75 points in total)
  - Item Image should be vertically centered inside the card - **0.25** points
  - The redirect icon should open the correct item on eBay – **0.25** points
  - All the fields as per the video should be present in the summary and detailed card with proper styling format – **1** point
  - The cards should have shadow effect on hover – **0.25** points
  - The item title should be truncated to 1 line for in the summary card and should be displayed completely for the detailed card. – **0.25** points
  - The item image should have similar zoom effect and animation as the video – **0.5** points
  - Default image should be shown in case there is no image available – **0.25** points
3. The headline displaying the total number of results below the search form should update correctly on search and during multiple searches as well – **0.25** points
4. The **Show More** / **Show Less** button functionality and the scroll effect should work as per the video – **0.5** points
5. In case of no results found, the message should display correctly – **0.25** points
6. Cards should be sorted as per the sorting criteria selected and displayed in the order they have been retrieved – **0.25** points
7. All the buttons (Clear, Search, Show More, Show Less) should have styling as shown in the video – **0.25** points

### Python script (2.5 points in total)

1. The API endpoint for search should return correct JSON data with all the required fields present (All the items that have any field missing should not be returned) – **0.5** points
2. The API endpoint for search should return only 10 valid items to the front-end – **0.5** points
3. Create an extra URL endpoint that needs to be added on the Homework link table for the filters – **0.5** points
  - Keyword: mask
  - Price range: 30\$ to 50\$
  - Condition: New
  - Shipping: Expedited
  - SortBy: PricePlusShippingHighestFirst
4. The JSON data should be parsed on the server and formatted as specified in the description document: 1 point

### Penalties:

1. **4-point penalty** for making direct calls from **JavaScript** to the eBay APIs.
2. **4-point penalty** for using frameworks like **Bootstrap** and **JQuery**.
3. **3-point penalty** if the Python server does not use the **Flask** framework.
4. If any students are found collaborating and / or sharing the same code, they will be reported to **Steve Bucher** at the Viterbi School of Engineering, and a formal academic violation report will be submitted.

### Source code submission:

Students **must** update their Homework #6 link in their coursework table to refer to the deployed homework-6 page. Additionally, students should save their source code. If there are concerns of collaboration or plagiarism, students will be requested to provide their source code for further evaluation.