

## **Lab #2 - ORF401 EL, SP, JJ**

Name: HopStar

### **4. Discuss how the JavaScript function checkForm could/should be made more generic (i.e., so that the same functions could be used with a variety of different HTML pages without having to be modified).**

We brainstormed a variety of ways in which checkForm could/should be made more generic as follows:

- Instead of using form.elements[0] to access the element in a form with a very specific index, we were viewing the HTML Form Object description and realized that form.elements['fieldname'] would be able to access specific data we want if inputs are not always in a specific order.
- Need a way to have flexible error messages. For example we just add a broad “Elon Musk” alert. What if someone puts numbers and or invalid characters like X-AE-12 for a name? We need a way to make alerts specifically point to which of our two inputs in the form has an error instead of it just saying “He’s not here”

### **5. Describe other ways in which JavaScript functions might be used to improve the functionality of HandyRides pages.**

Javascript can enhance functionality and user experience of HandyRides, HopStar, or any ride-sharing service by complimenting HTML → One (HTML) handles the way a website looks, the other (Javascript) handles the way a website operates, processes data, and returns information to the users. Javascript can handle the real-time data that is used to adjust ride updates without having to refresh the page, can use mapping APIs to display routes and calculate any relevant data in those routes, etc all while HTML displays. From this Lab, we realized how sort of the “back-end” of Javascript is able to perform the complex functions that go into operating a website that is very relevant to user retention by controlling cookies, providing alerts, and account management for the users which are expected in any modern website.

### **6. Discuss the advantages and disadvantages of saving state information (e.g. information about previous usage) on the client versus saving it on the server.**

Storing state information on the client produces reduced server requests, database load, and is able to enhance website performance while making it so that running the database is cheaper with less data to hold. In other words, it conserves server resources and improves responsiveness by storing data on the client. However, very notable disadvantages include vulnerability of client data to unauthorized alteration or access (shoutout to COS217 Buffer Overrun Attack assignment), contrasted to high security standards in server-side storage. Additionally, the sever holds less information about the user and thus is not able to provide a personalized experience,

use user behavior to determine important analytics, and retention might be lower with less frequent strategic decisions on what we show users.