**READ ME-**

**I have created and tested the app using XAMPP server (localhost).**

**In this app, I have created the front-end (index.html) as per the requirements and the JSON files are hardcoded (emulating submissions to the fake API) considering the GET and POST requests will be send to a backend in real-time scenario and the backend will return the response in JSON format. The data from these responses will be displayed to the user. Any manipulation needed to this data is shown in the app.js file.**

**Three different cases have been emulated here based on the given rules:**

1. **A Queen room reservation where user selects 1 offer to upgrade to, say, a King room.**

***Relevant JSON files*: reservation1.json (emulating the GET request made to fetch Queen room details and display it), reservation1\_confirm.json (emulating the response to POST request after an offer is selected and submitted to the fake API)**

1. **A King room reservation where user selects 2 offers to upgrade to a Premium Suite with Champagne/Strawberry.**

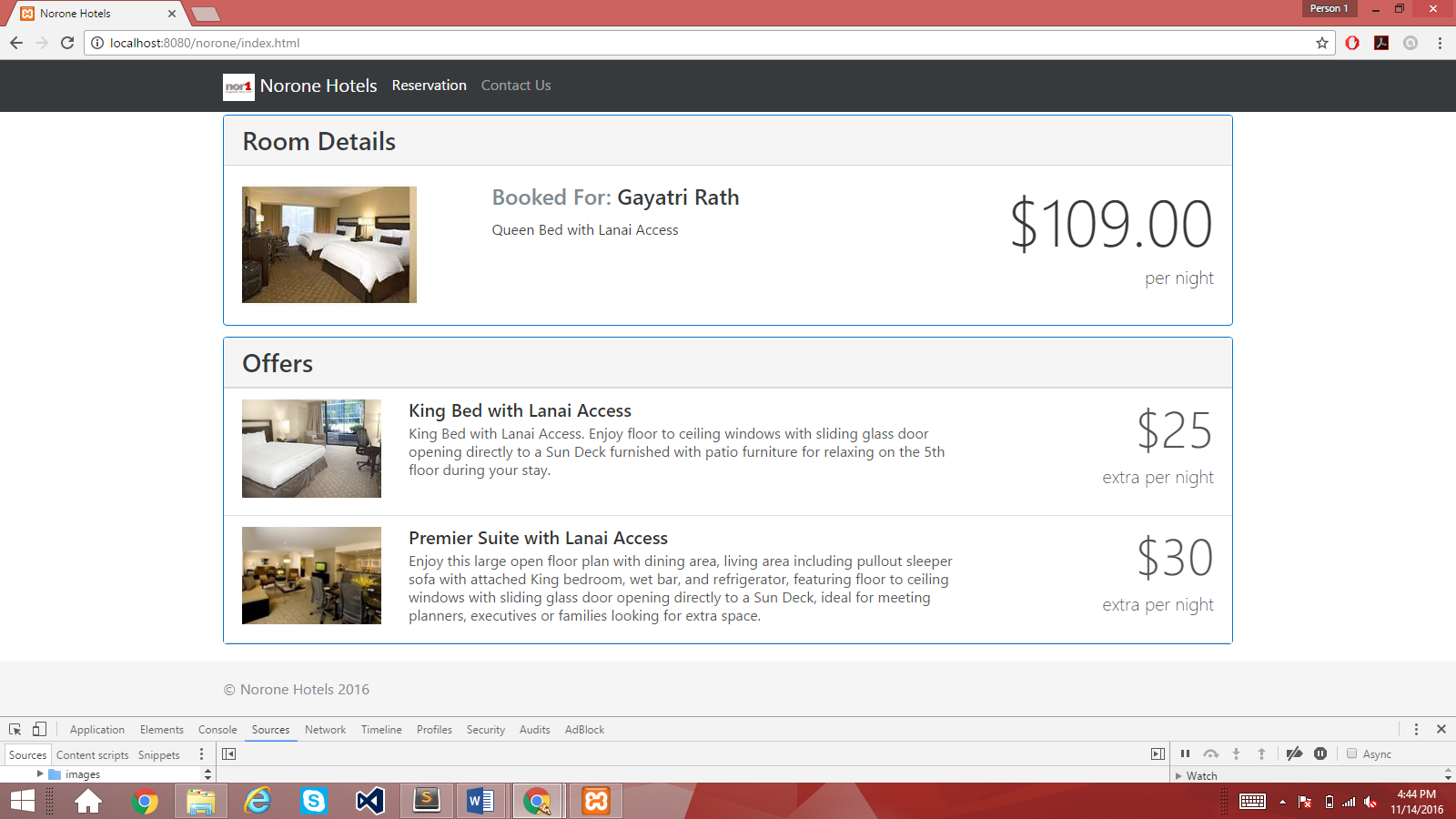
***Relevant JSON files*: reservation2.json (emulating the GET request made to fetch King room details and display it), reservation2\_confirm.json (emulating the response to POST request after multiple offers are selected and submitted to the fake API)**

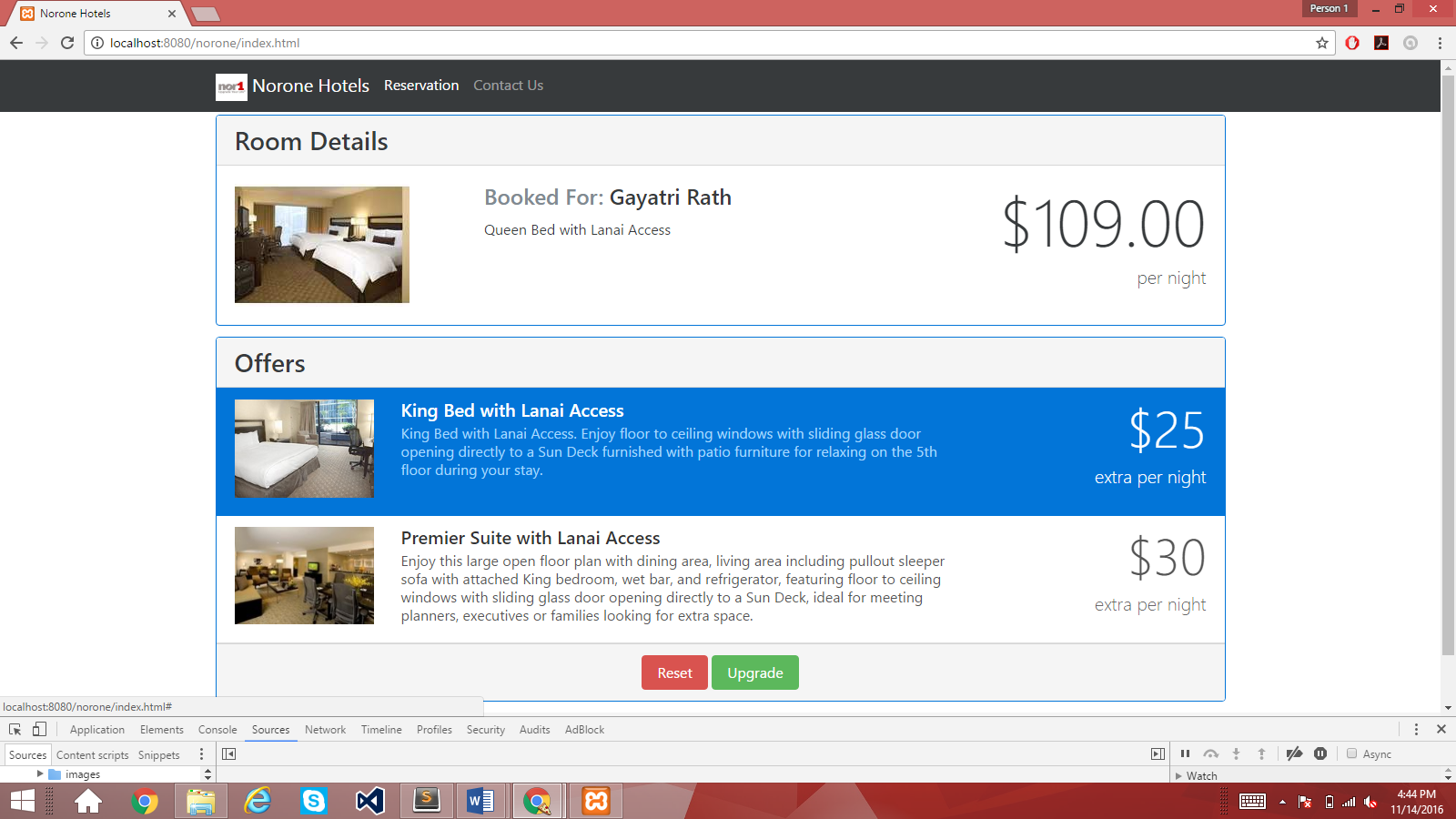
1. **A Premium Suite reservation where no offers are available for the user to select.**

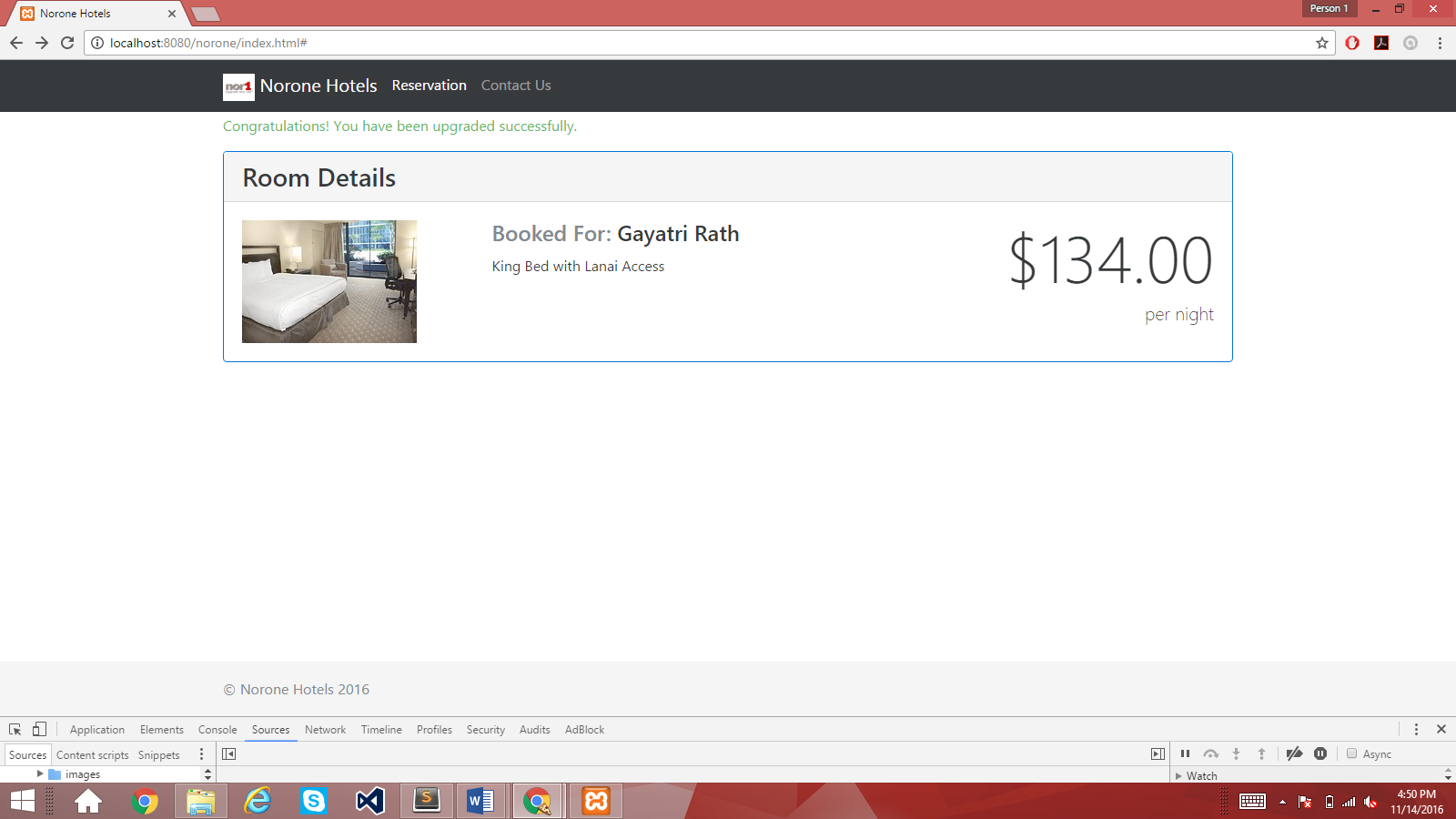
***Relevant JSON files*: reservation3.json (emulating the GET request made to fetch Premium Suite details and display it).**

**Sample Screenshots:**

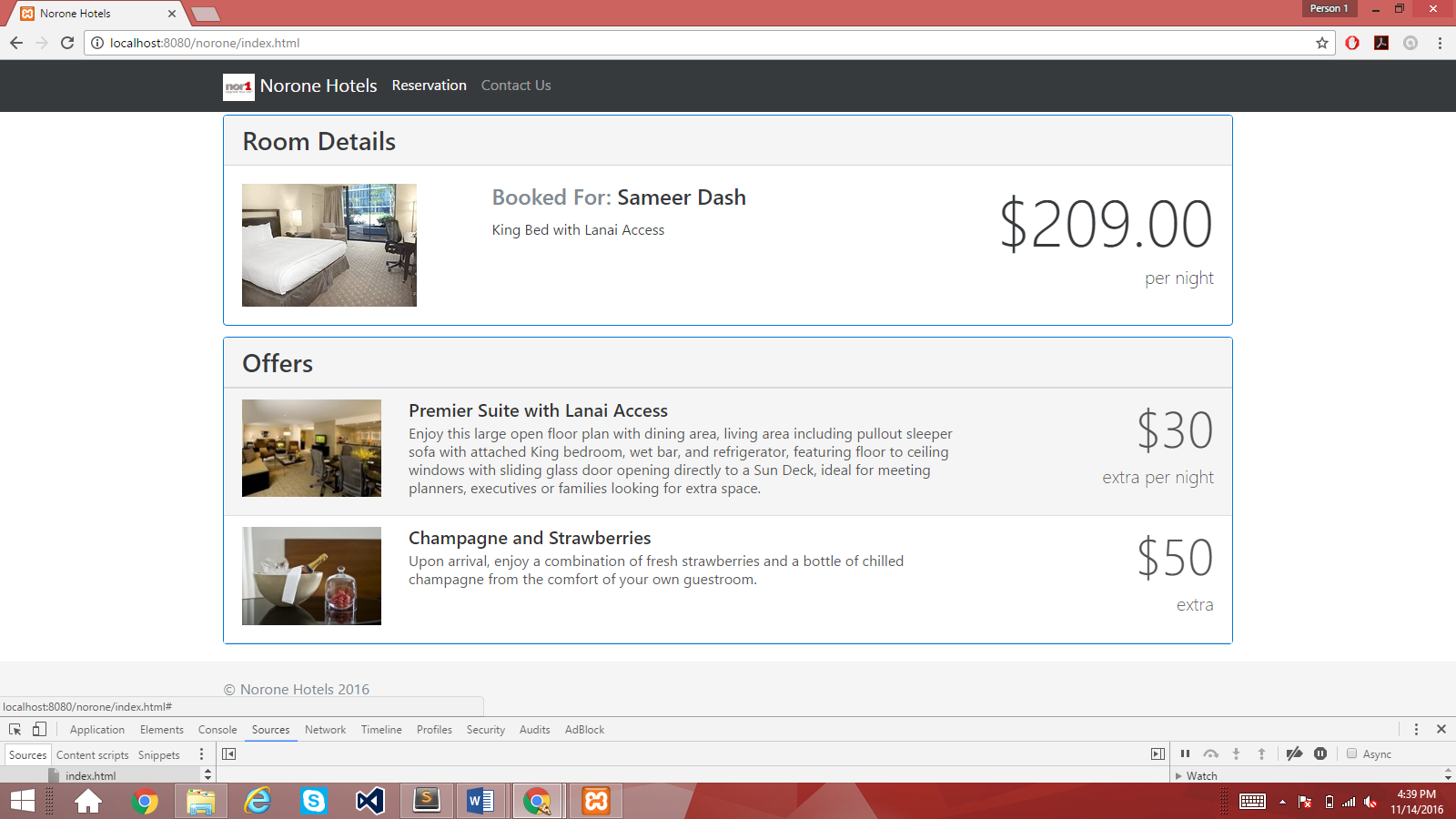
**1) Queen room reservation upgraded to King room**

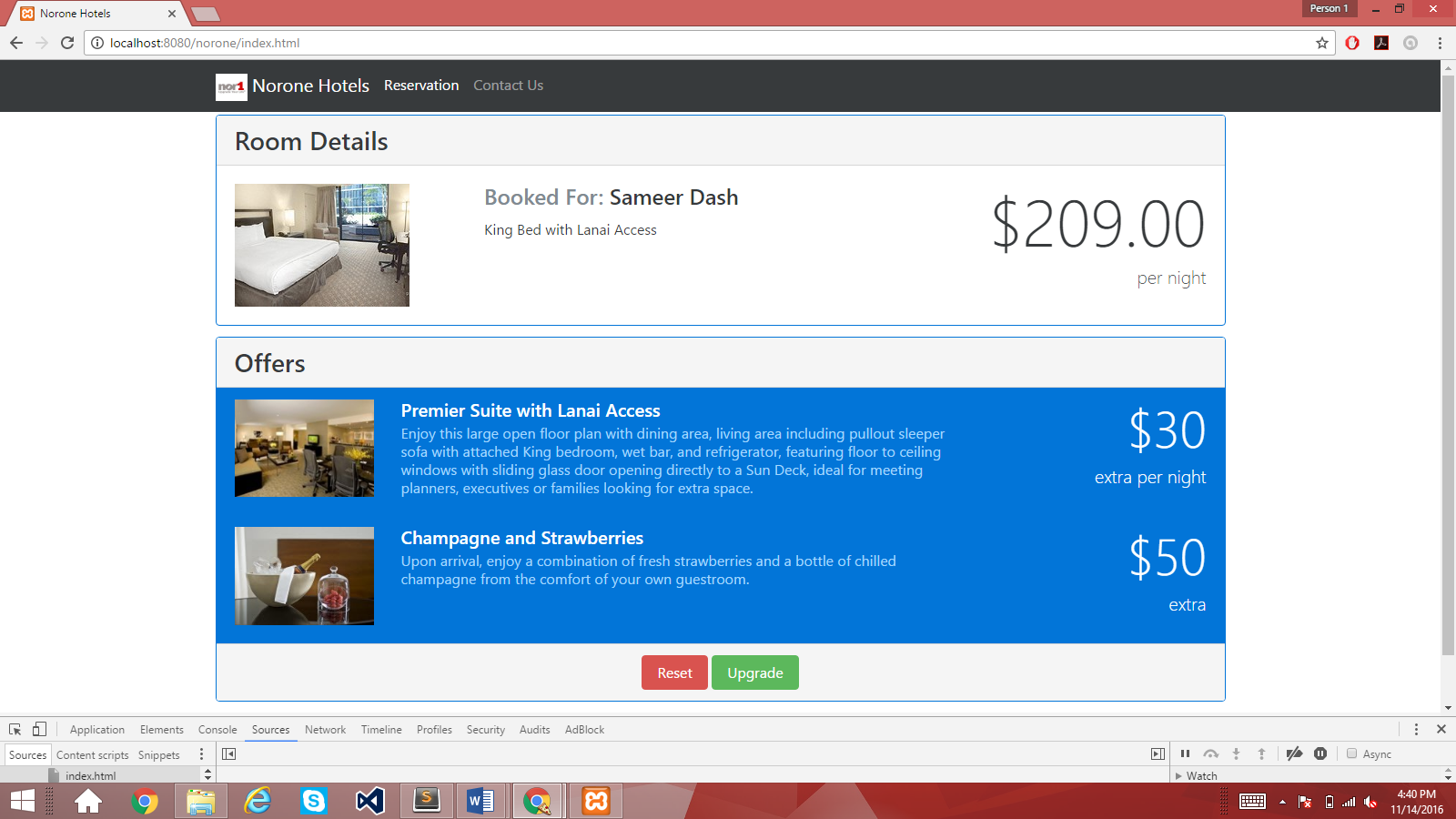


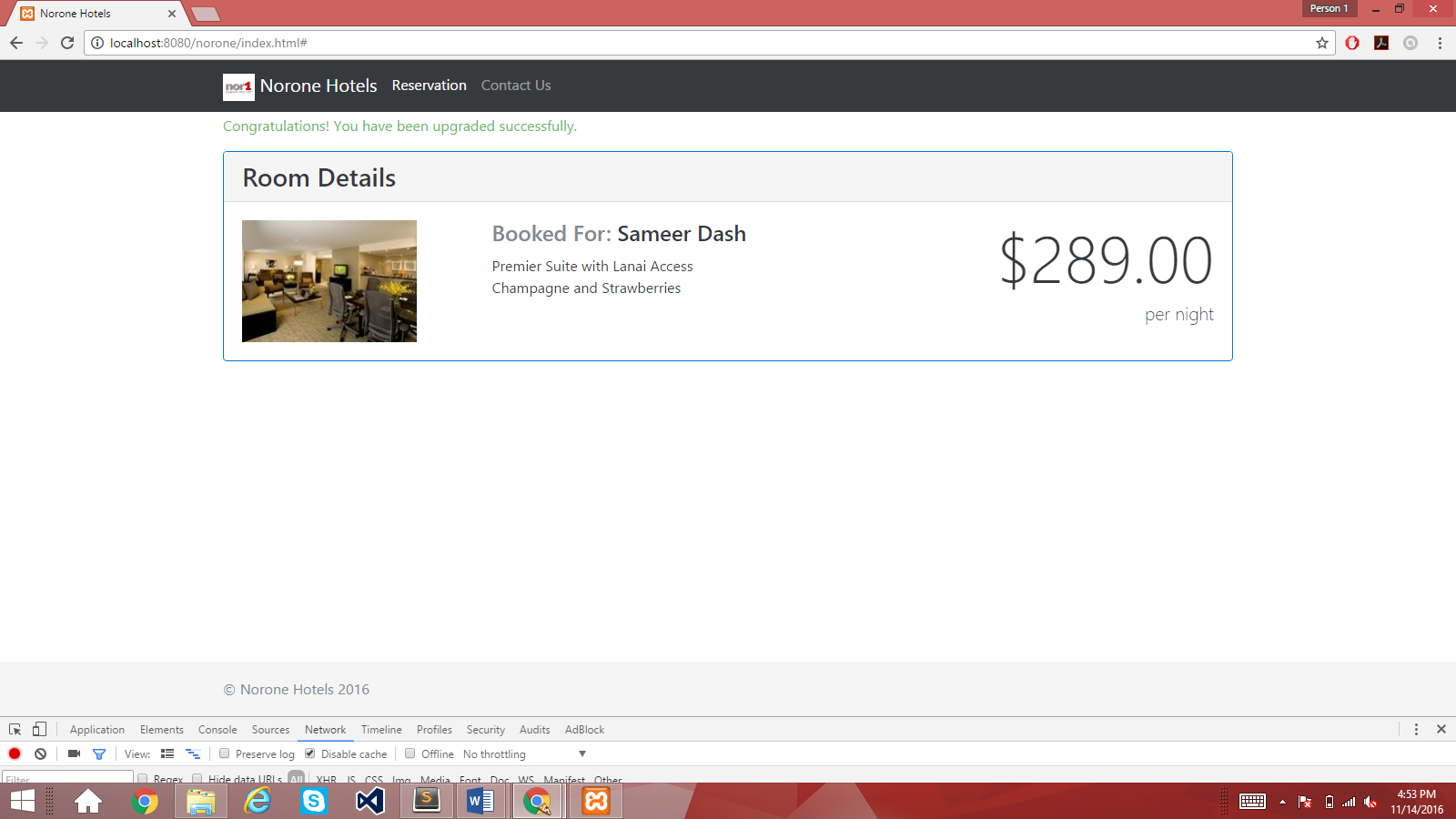




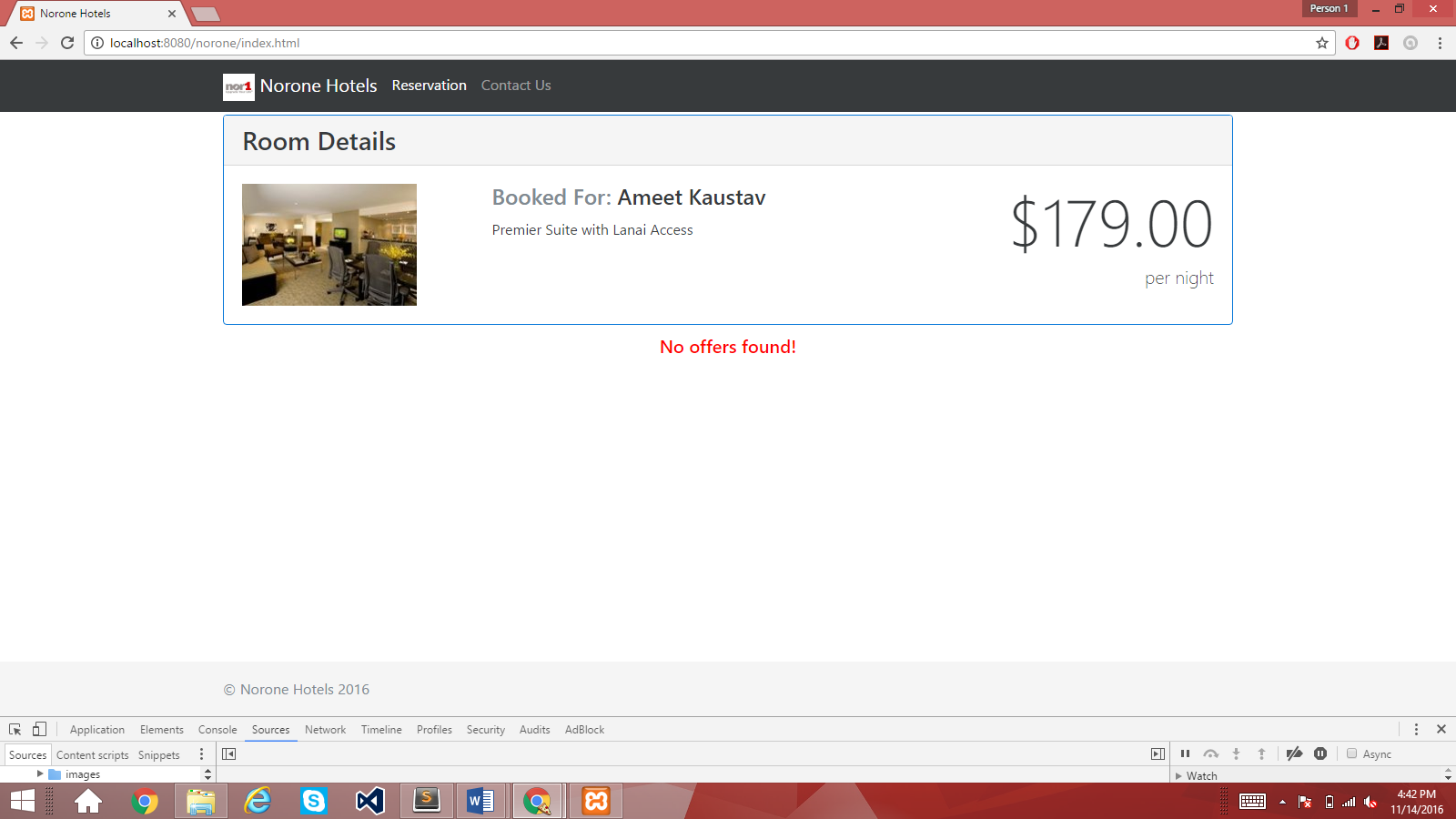
**2) King room reservation upgraded to Premium suite w/ Champagne and Strawberry**







**3) Premium Suite reservation with no offers available to choose from**



**Other Questions:**

**Q: How would you handle no offers available if there are no matching rules?**

**Ans: Shown in code (case 3)**

**Q: How would you handle branding and theming for different hotels?**

**Ans: For different hotels, branding can be handled by displaying the appropriate logo, contact information, relevant offers, appropriate navigation tabs. The offers available and the rules for these offers will be different for different hotels. The logo, nav bar links, contact info can be changed in the <nav> element of the HTML file. In this app, I have used bootstrap 4 which allows easy creation of various themes which can be usd for different hotels.**

**Q: How should 10+ different languages be supported?**

**Ans: For multi-language support, I would**

* **Create a dropdown list with multiple languages to select from and in javascript I would have a look-up array with corresponding language codes.**
* **Using ng-model, I would bind the user’s selection and find the appropriate language code from the look-up array.**
* **Suppose, the language code found from the look-up array is stored in a $scope.lang variable.**
* **Then I would modify the default http request headers by accessing angular js $http service, ‘**$httpProvider.defaults.headers’ object. The GET and POST request header’s‘accept-language’ property can be then assigned to the language code in $scope.lang.
* This would give the response back in that language which can be displayed to the user.
* If there are multiple pages through which we need to retain the same language selected by the user, we can user local storage to store the user’s selection.

**Q: How should 10+ different currencies be supported?**

**Ans: One way is, Angular JS has built-in internationalization for the currency filter. According to the locale of the country, its corresponding js can be added which can show the currency in that particular format.**

**Another way would be to use a 3rd party currency converter provider which can convert the currency value from one currency to another based on current exchange rates.**

* **We can have a currency dropdown list with multiple currencies to select from.**
* **When the user selects a particular currency, a call to the provider can return the equivalent value in the chosen currency which can be displayed to the user.**
* **There are ASCII codes available for displaying currency symbols.**