**Student ID:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Full Names:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Web Application Programming

(CS472)

(40 points)

(June 2022)

Instructor: O. Kalu

Sample Midterm Exam

1. The exam duration is 1 hour 30 minutes.
2. The exam is an online, computer-based exam; so, you may use a computer for both the Part 1 (theory) and Part 2 (coding) tasks.
3. **This exam paper document is a copyrighted material and so it must not be copied or reproduced or transferred or shared or distributed**.
4. You are expected to type all your answers for Part 1, as text into this document. You may also use an IDE or any Code Editor tool of your choice to implement your solutions for the questions in the Part 2 (Web Application Coding).
5. Upon completion, put your entire Exam (including the projects/folders with your **source code** and this document with your typed-in answers (in either Microsoft Word or Adobe PDF format, only)**)** into a single zip file named **MidtermExam.zip**, and submit to Sakai, under the Assignment titled, “Midterm Exam”.
6. **NOTE**: ***If you fail to submit your exam to Sakai because it has past the Submission due time, and you then email it instead, then be aware that your maximum possible score will be 80%,and only if your work scores up to that level.***

--------------------------------------------------------------------------------------------------------------------

Make sure to include the screenshots of your results, where it is required.

--------------------------------------------------------------------------------------------------------------------

(CS472 - WAP)

Part 2 – WebApp Coding (60 points)

**Part II – JavaScript/Web Application Programming skills:** (60 points)

**Note:** *For the tasks in these questions, where applicable, you are expected to take screenshot(s) of your web UI(s), save into a .png or .jpg image file, placed inside a folder named, screenshots and include these in the MidtermExam.zip file, you submit or you may simply copy/paste your screenshots to the bottom of the associated question(s) right here in this document.*

1. (30 points) **Implementing a client-side JavaScript Web Application**

Assume you have been hired by the Regional Office of Census Administration of the South West states, USA. And they have tasked you to design and build a website for the 2020 Census, including just a homepage and a Census Data Collection form (see the sample web UIs shown below).

Using HTML, CSS and JavaScript, implement the website, as shown in the UI screenshots below, with the features and functionalities, specified/listed further below:

Homepage:

Graphical user interface, text, application, Word

Description automatically generated

Census Form page:

Graphical user interface, text, application

Description automatically generated

1.1 Code the User interface of the web application using standards-compliant, semantically-correct HTML5 markup, including all the web pages and form fields as shown in the UI screenshots above.

1.2 Apply styling using Bootstrap or you may apply your own custom CSS styling to produce the same or similar look and layout. Note: Your UI does NOT necessarily have to be exactly the same as the sample shown above. But it should have all the necessary form data input fields, labels, images and buttons etc.

1.3 Citizen ID, Social Security Number, Full Name, State and Senior Citizen are all required data, to submit the Census form.

1.4 Add validation using appropriate regular expression to ensure that each Citizen ID and Social Security Number entered must be in the specified format of, XXXXXXXXXX and XXX-XX-XXXX, respectively, as shown on the Census form UI above. And where X is a numeric digit.

1.5 The user can record citizens’ data for the following 3 states of the South West region – Arizona, New Mexico and Texas. Therefore, your drop-down list data field on the Census form should contain these 3 options.

1.6 Add JavaScript code to implement a feature such that when the Submit button is clicked and the Census Form data is submitted, the web browser should display the data entered in an Alert popup dialog box, like shown below.

Graphical user interface

Description automatically generated

(Please note: Points will be awarded based on your adherence to Web programming recommended best practices such as use of standards-compliant, semantically-correct HTML5 markup, unobtrusive CSS, unobtrusive JavaScript etc).

1. (30 points) **Implementing a client-side JavaScript Web Application**

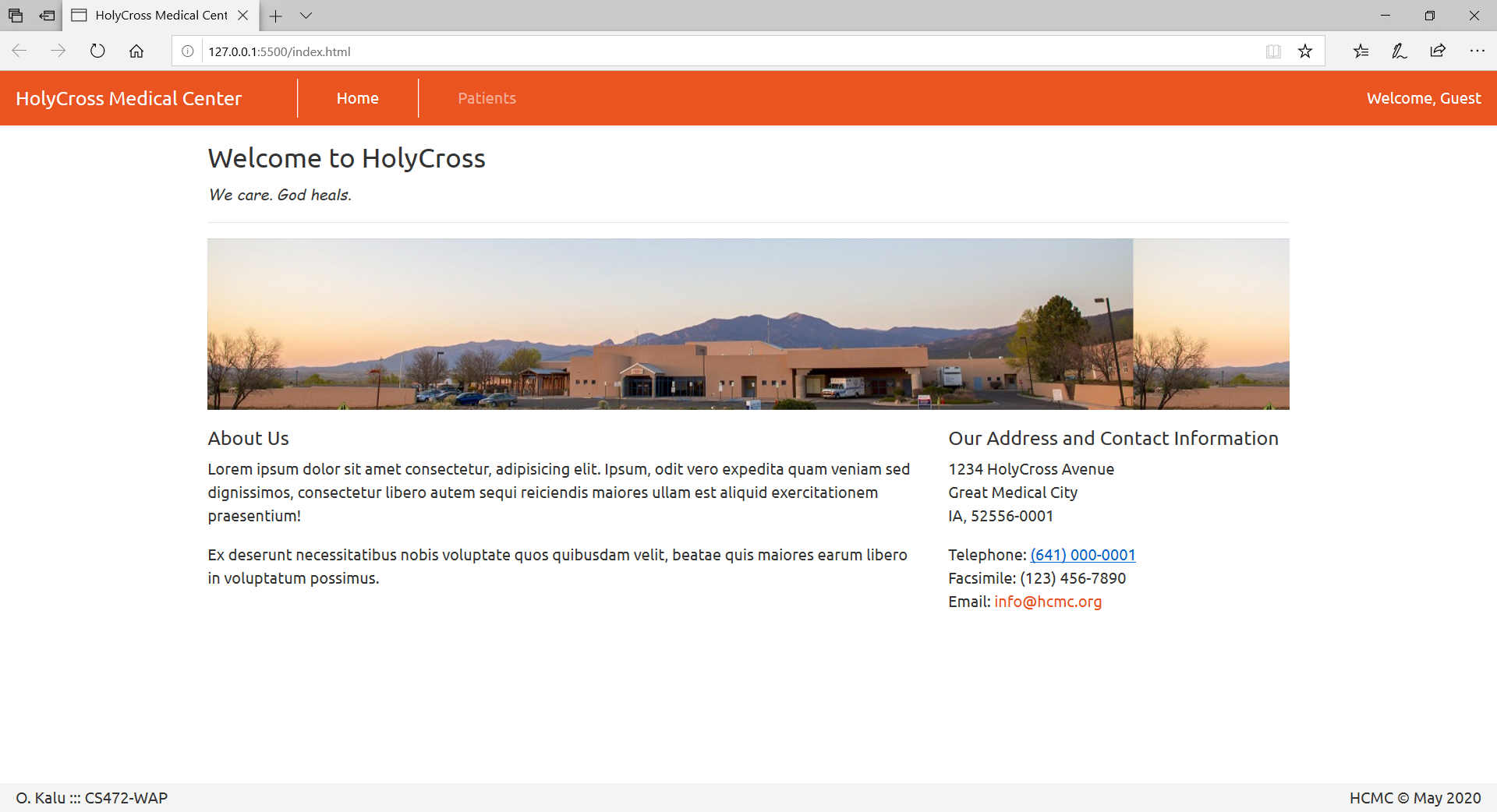
A popular city hospital, named HolyCross Medical Center (HCMC), has hired you to develop a web-based system for them, which they will be using to run part of their hospital operations. Specifically, the system will be used for registering their **Patient**s. Especially important to the Chief Medical Director of HCMC is, the data that provides information about their Elderly Patients and Out-patients.

**An Elderly Patient is any patient who is of age, 65 or older.**

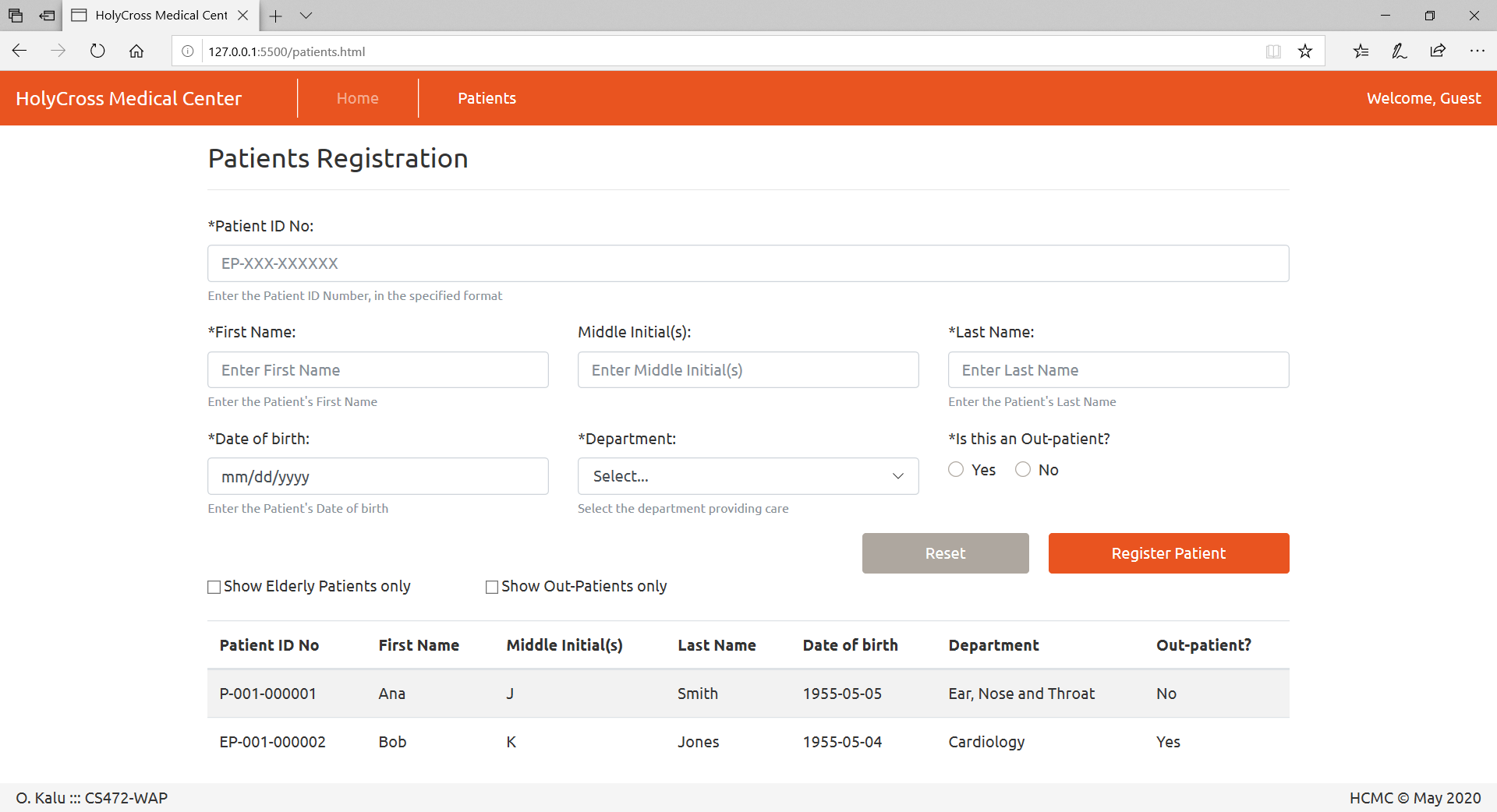
The website you are asked to build should have just a Homepage and a Patients Registration Form page (see the sample web UIs shown below).

Using HTML, CSS and JavaScript, implement the website, as shown in the UI screenshots below, with the features and functionalities, specified/listed further below:

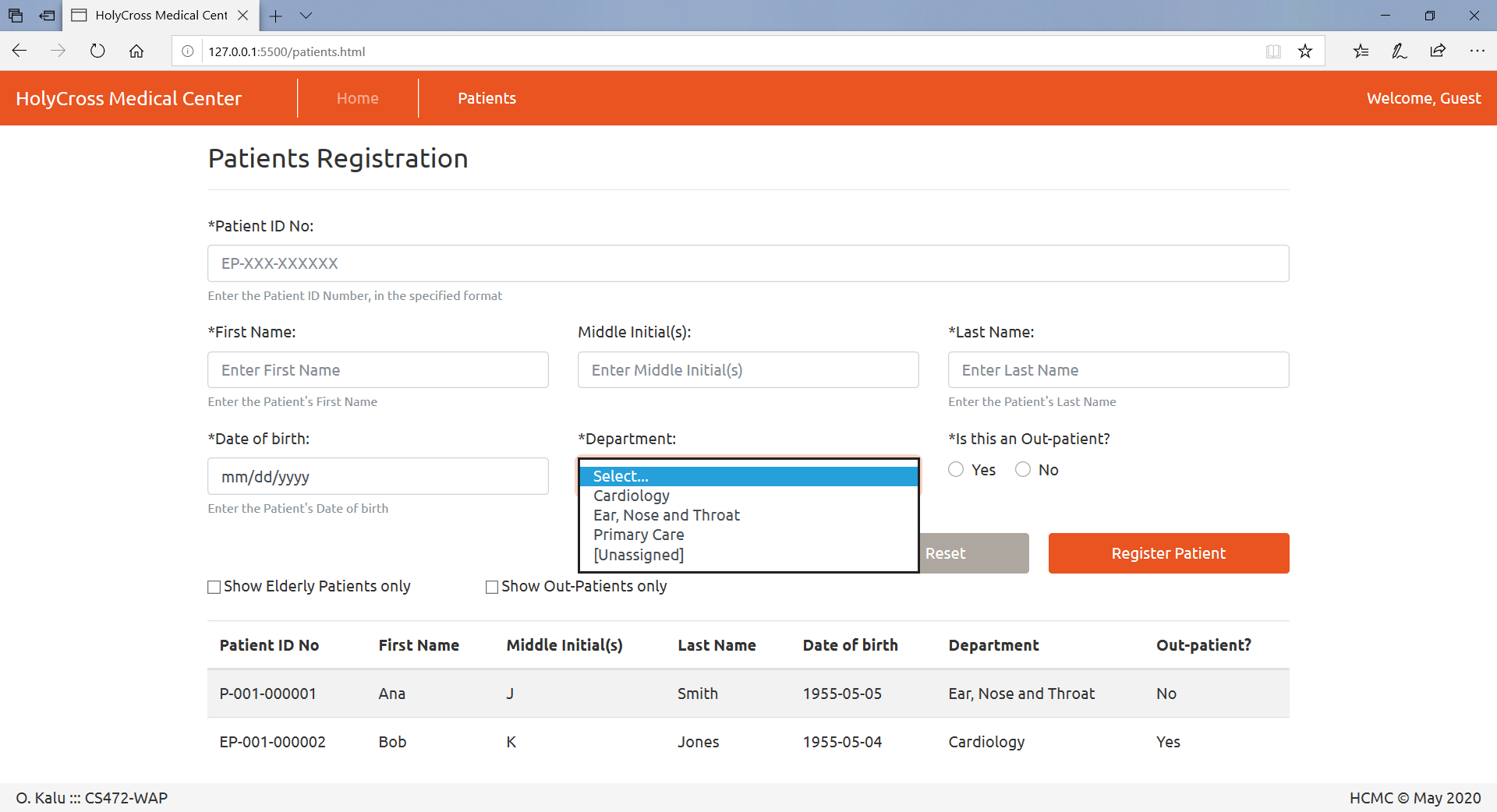
**Homepage**:



**Patients Registration Form page**:



The data inside the Department drop-down list:



* 1. Code the User interface of the web application using standards-compliant, semantically correct HTML5 markup, including both the web pages and form fields as shown in the UI screenshots above.
  2. Apply styling using Bootstrap or your own custom CSS styling, to produce the same or similar look and layout. **Note:** Your UI does NOT necessarily have to be exactly the same as the sample shown above. But it should have all the necessary form data input fields, labels, images and buttons etc.
  3. Patient ID Number, First Name, Last Name, Date of birth, Department and Out-patient are all required data, to register a Patient into the system.
  4. Add data validation check using appropriate regular expression to ensure that each Patient ID Number entered must be in the specified format of, either P-XXX-XXXXXX or EP-XXX-XXXXXX, as shown on the Patient Registration form above. And where X is a numeric digit.
  5. Using JavaScript and the DOM API, implement code to register the 2 sample patients’ data listed below, which also appends/displays the data in a table at the bottom of the page, when the “Register Patient” button is clicked and the form is submitted, as shown in the screenshot above. The user should be able to register many patients’ data and see them listed on the webpage.

**Sample Patients’ Data for submission to test/demonstrate your web application**:

Patient 1:

Patient ID Number: P-001-000001,

First Name: Ana,

Middle Initial(s): J,

Last Name: Smith,

Date of birth: 5 May 1955

Department: Ear, Nose and Throat,

Is out-patient?: No

Patient 2:

Patient ID Number: P-001-000002,

First Name: Bob,

Middle Initial(s): K,

Last Name: Jones,

Date of birth: 4 May 1955

Department: Cardiology,

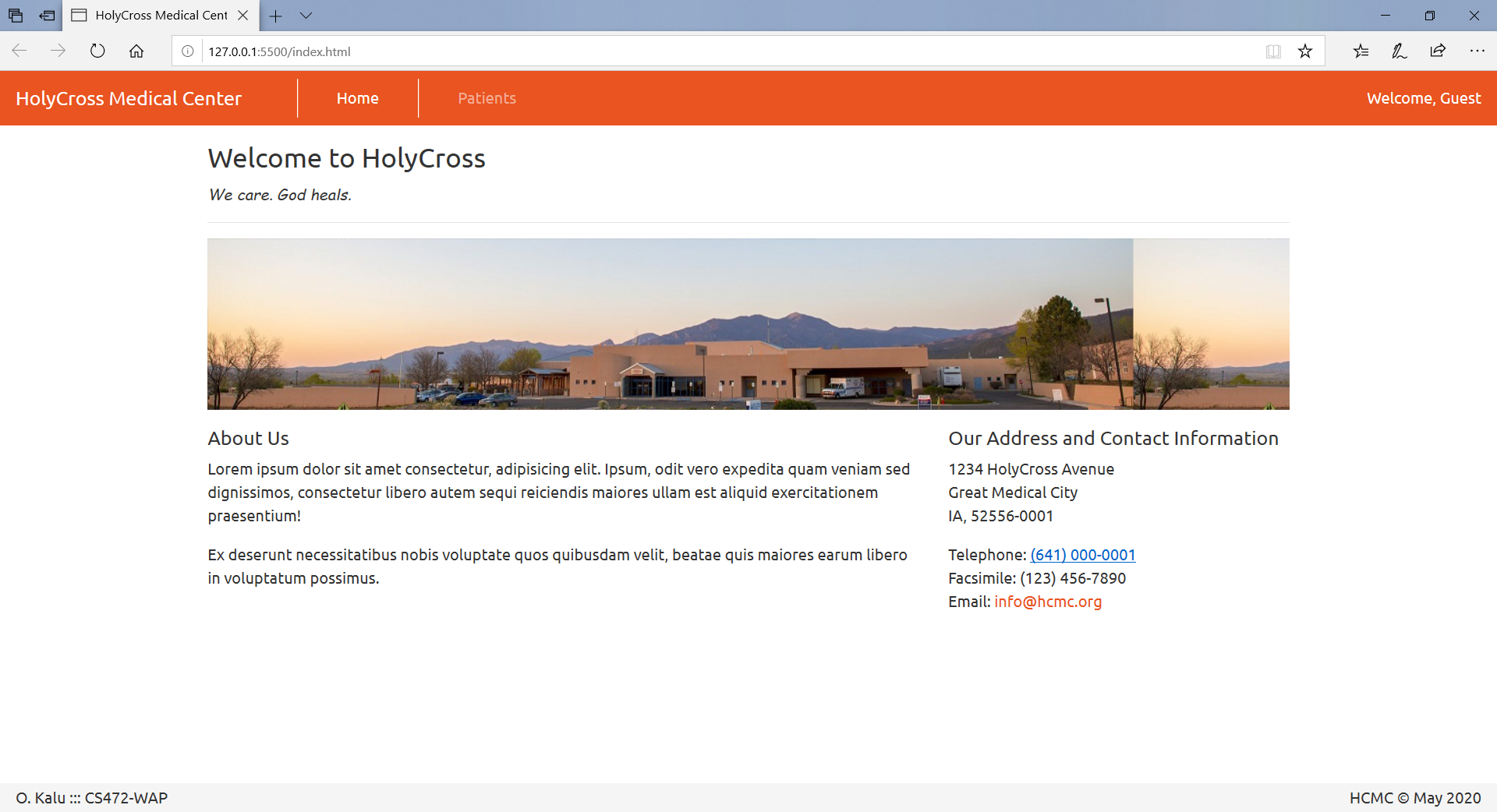
Is out-patient: Yes

* 1. Implement code for just ONE of the following two optional features:
     1. Implement code such that when the user selects/checks the checkbox labelled, “Show Elderly Patients only”, the data table will display only data for the Elderly Patients. And when the user deselects/unchecks the checkbox labelled, “Show Elderly Patients only”, the data table will re-display all the Patients data again. **Note**: The criterion for determining which Patient is considered an Elderly Patient is given above.

OR

* + 1. Implement code such that when the use selects/checks the checkbox labelled, “Show Out-Patients only”, the data table will display only data for the Out-Patients. And when the user deselects/unchecks the checkbox labelled, “Show Out-Patients only”, the data table will re-display all the Patients data again.

(**Please note:** Points will be awarded based on your adherence to Web programming recommended best practices such as use of standards-compliant, semantically-correct HTML5 markup, unobtrusive CSS, unobtrusive JavaScript etc).



**//-- The End --//**