**Assignment -3**

Cloud object storage and Watson Assistant

|  |  |
| --- | --- |
| Assignment Date | 19 September 2022 |
| Student Name | DEVANAND V |
| Student Roll Number | 7376191CS151 |
| Maximum Marks | 2 Marks |

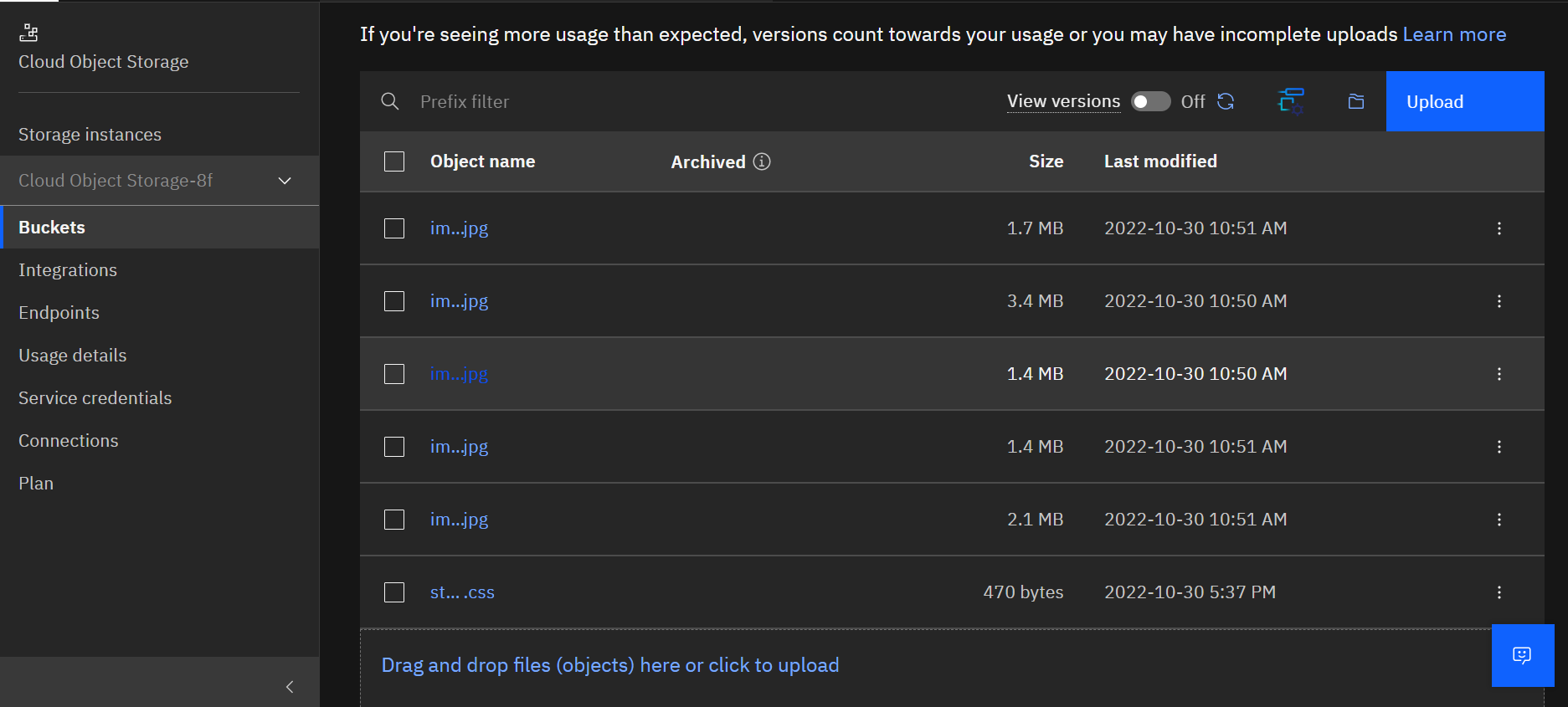
**Question-1:**

1.Create a Bucket in IBM object storage.

2.Upload an 5 images to ibm object storage and make it public. write html code to displaying all the 5 images.

3.Upload a css page to the object storage and use the same page in your HTML code.

**Bucket image:**



**object\_config.py**

import ibm\_boto3

from ibm\_botocore.client import Config, ClientError

# Constants for IBM COS values

COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints

COS\_API\_KEY\_ID = "Bc32PN-Ec2mpFRVtchtdfkBJBM-uDrjwYdY12L0knCH4" # eg "W00YixxxxxxxxxxMB-odB-2ySfTrFBIQQWanc--P3byk"

COS\_INSTANCE\_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/fc99842817e447e4be38d2cca7268d9c:183cba6d-ad4f-4d17-b129-61f93943e367::"

# Create resource

cos = ibm\_boto3.resource("s3",

    ibm\_api\_key\_id=COS\_API\_KEY\_ID,

    ibm\_service\_instance\_id=COS\_INSTANCE\_CRN,

    config=Config(signature\_version="oauth"),

    endpoint\_url=COS\_ENDPOINT

)

def get\_buckets():

    print("Retrieving list of buckets")

    try:

        buckets = cos.buckets.all()

        for bucket in buckets:

            # print("Bucket Name: {0}".format(bucket.name))

            return bucket.name

    except ClientError as be:

        print("CLIENT ERROR: {0}\n".format(be))

    except Exception as e:

        print("Unable to retrieve list buckets: {0}".format(e))

bucket\_name = get\_buckets()

print(bucket\_name)

def get\_bucket\_contents(bucket\_name):

    print("Retrieving bucket contents from: {0}".format(bucket\_name))

    try:

        file\_names = []

        files = cos.Bucket(bucket\_name).objects.all()

        for file in files:

            # print("Item: {0} ({1} bytes).".format(file.key, file.size))

            if 'img' in file.key:

                file\_names.append(file.key)

        return file\_names

    except ClientError as be:

        print("CLIENT ERROR: {0}\n".format(be))

    except Exception as e:

        print("Unable to retrieve bucket contents: {0}".format(e))

object\_files = get\_bucket\_contents(bucket\_name)

**app.py:**

from flask import Flask, render\_template, redirect

from object\_config import object\_files

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

    return render\_template('index.html', files = object\_files)

if \_\_name\_\_=='\_\_main\_\_':

    app.run()

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://s3.jp-tok.cloud-object-storage.appdomain.cloud/obj-bucket-1/styles.css">

    <!-- <link rel="stylesheet" href="{{ url\_for('static',filename='css/styles.css') }}"> -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

        crossorigin="anonymous"></script>

    <title>IBM Object Storage</title>

</head>

<body>

    <div class="content">

        <h1 class="display-1">Object Storage images</h1>

        <div class="row">

            {% for img in files: %}

            <div class="col">

                <div class="polaroid">

                    <img src="https://s3.jp-tok.cloud-object-storage.appdomain.cloud/obj-bucket-1/{{img}}" alt="{{img}}"

                        style="width:100%">

                    <div class="label-text">

                        <p>{{img}}</p>

                    </div>

                </div>

            </div>

            {%endfor%}

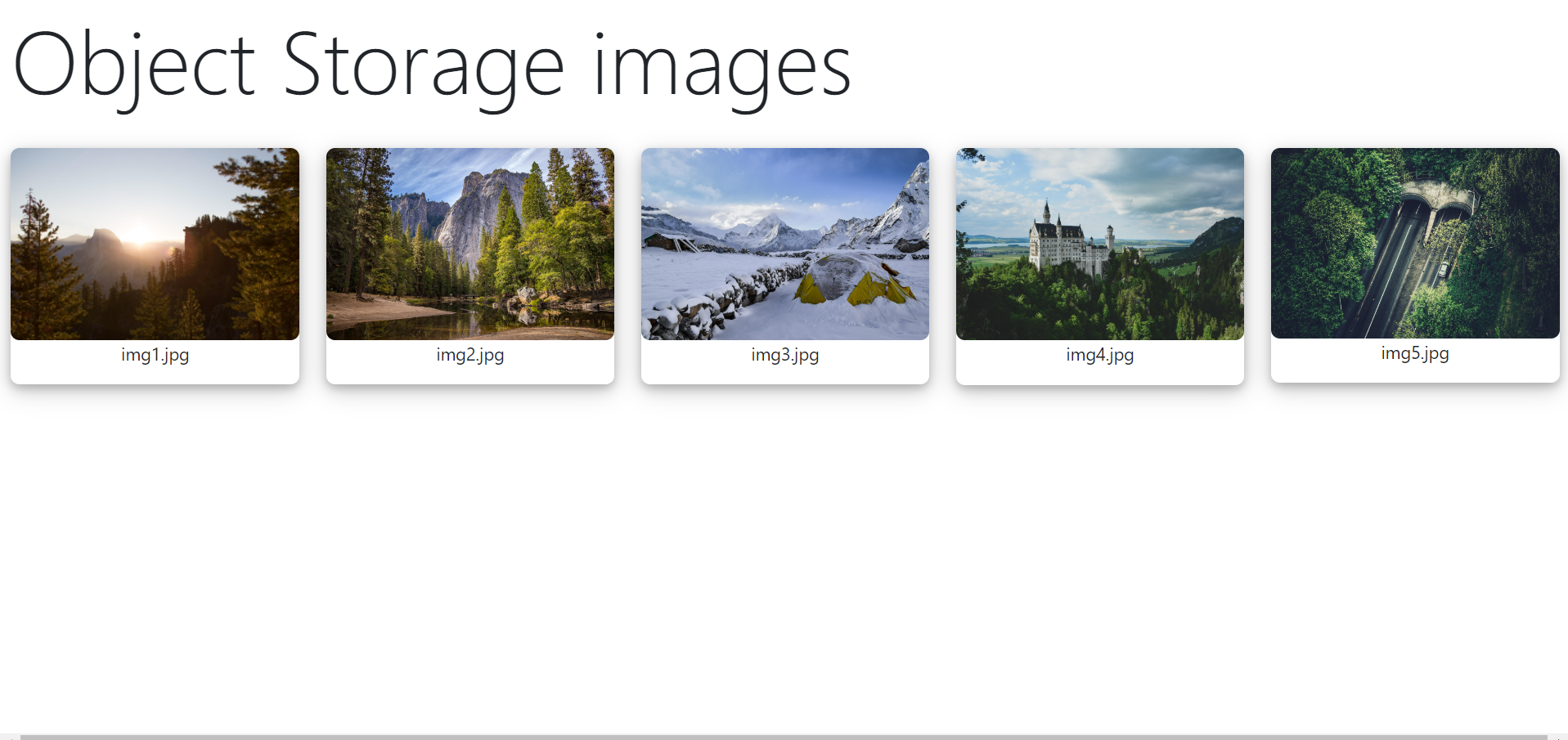
        </div>

    </div>

</body>

</html>

**Final output:**

****

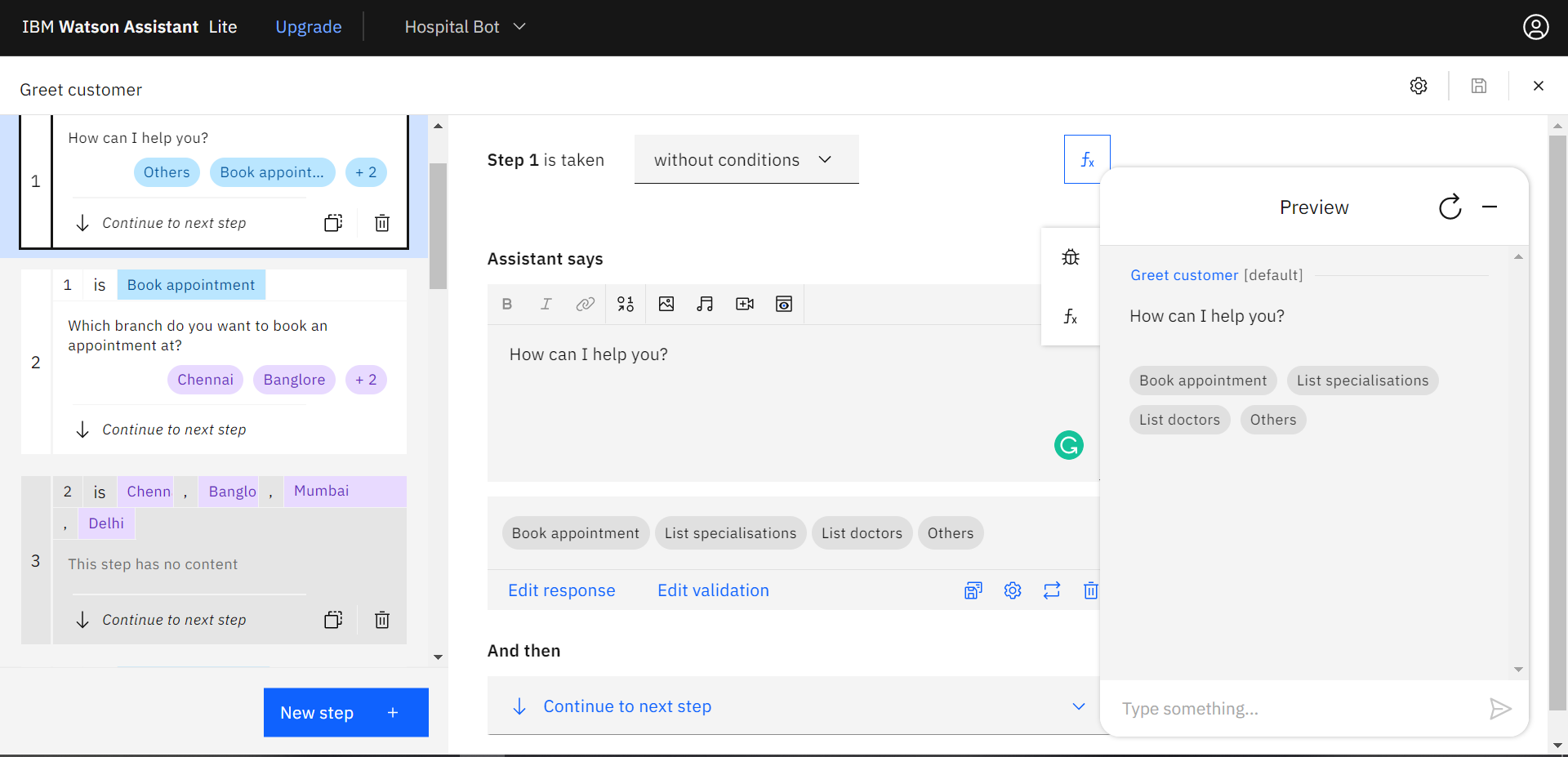
4.Design a chatbot using IBM Watson assistant for hospital. Ex: User comes with query to know the branches for that hospital in your city. Submit the web URL of that chat bot as a assignment.

5.Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.

**Url**

[**https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-cb7201bb-9098-466d-9b33-f39a179557bf%3A%3A9042e148-f016-4634-a34d-1fc3202e7a68&integrationID=3e7b6047-e838-477d-adbe-4a28e573cc27&region=au-syd&serviceInstanceID=cb7201bb-9098-466d-9b33-f39a179557bf**](https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-cb7201bb-9098-466d-9b33-f39a179557bf%3A%3A9042e148-f016-4634-a34d-1fc3202e7a68&integrationID=3e7b6047-e838-477d-adbe-4a28e573cc27&region=au-syd&serviceInstanceID=cb7201bb-9098-466d-9b33-f39a179557bf)

**Steps and condition:**



**Script for chatbot:**

 <script>

        window.watsonAssistantChatOptions = {

          integrationID: "3e7b6047-e838-477d-adbe-4a28e573cc27",

          region: "au-syd",

          serviceInstanceID: "cb7201bb-9098-466d-9b33-f39a179557bf",

          onLoad: function(instance) { instance.render(); }

        };

        setTimeout(function(){

          const t=document.createElement('script');

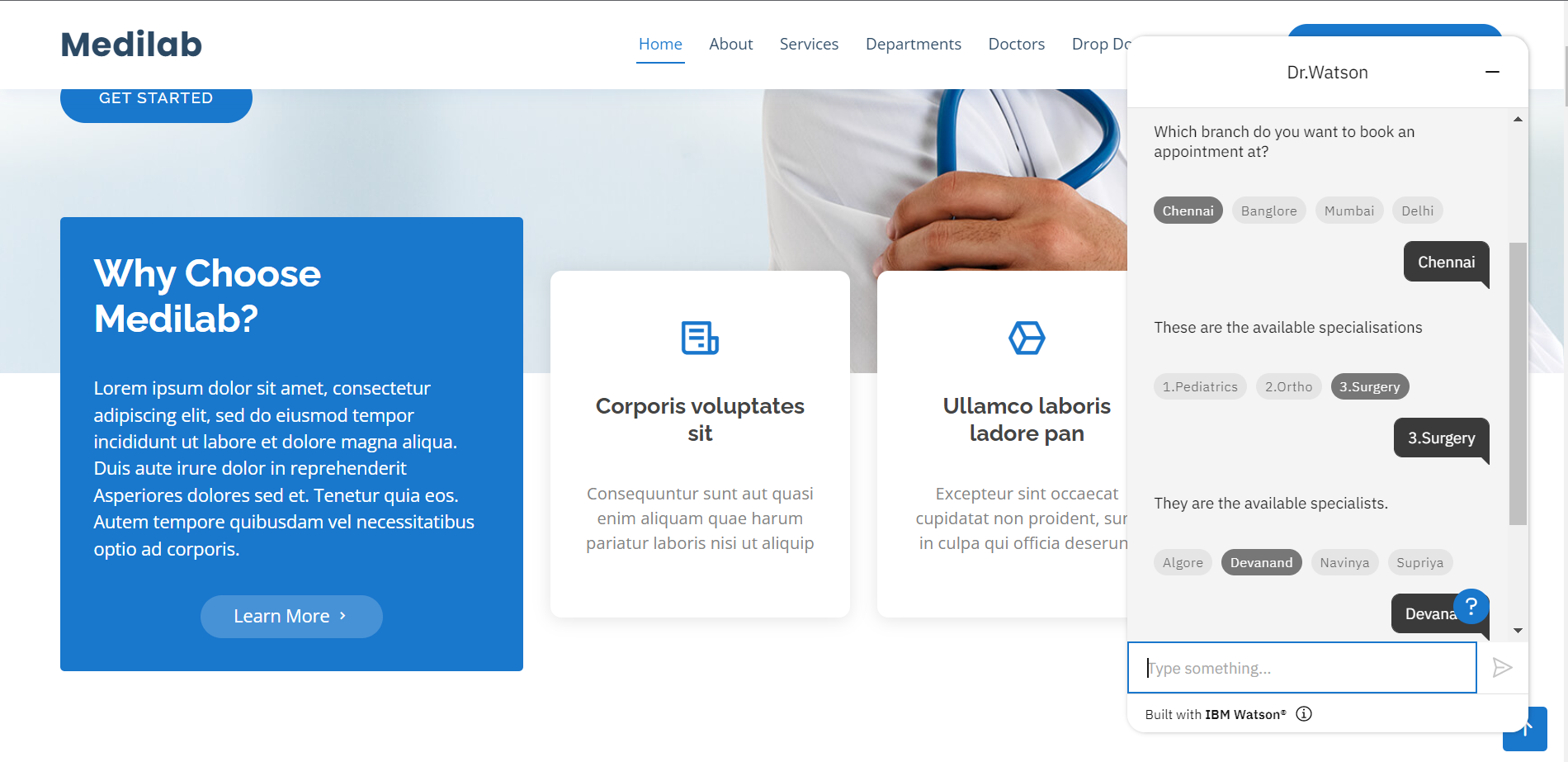
          t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

          document.head.appendChild(t);

        });

      </script>

**HTML output:**

****