Assignment -3

1. **Create a Bucket in IBM object storage.**

Request:

```
PUT /images HTTP/1.1
Content-Type: text/plain
Host: 67.228.254.193
X-Amz-Date: 20160821T052842Z
```

Response:

```
 HTTP/1.1 200 OK
Date: Wed, 24 Aug 2016 17:45:25 GMT
X-Clv-Request-Id: dca204eb-72b5-4e2a-a142-808d2a5c2a87
Accept-Ranges: bytes
Server: Cleversafe/3.9.0.115
X-Clv-S3-Version: 2.5
x-amz-request-id: dca204eb-72b5-4e2a-a142-808d2a5c2a87
Content-Length: 0
```

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

Request

```
POST /example-bucket HTTP/1.1
Authorization: {authorization-string}
x-amz-date: 20160825T183001Z
x-amz-content-sha256: 309721641329cf441f3fa16ef996cf24a2505f91be3e752ac9411688e3435429
x-amz-tagging: tag1=value1&tag2=value2
Content-Type: text/plain; charset=utf-8
```

```
Host: 67.228.254.193

Content-Type: multipart/form-data; boundary=---------------------------2393619733680

Content-Length: 639


Response

-----------------------------2393619733680

Content-Disposition: form-data; name="key"

uploads/${filename}

----------------------------2393619733680

Content-Disposition: form-data; name="Content-Type"

text/plain

----------------------------2393619733680

Content-Disposition: form-data; name="file"; filename="queen-bee.txt"

Content-Type: text/plain

The 'queen' bee is developed from larvae selected by worker bees and fed a

 substance referred to as 'royal jelly'. After a short while the 'queen' is

 the mother of nearly every bee in the hive, and the colony will fight

 fiercely to protect her.

----------------------------2393619733680--
```

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

App.js

```
var express = require('express');
var cfenv = require('cfenv');
var bodyParser = require('body-parser');
var app = express();
//...
```

```
// start server on the specified port and binding host
var port = process.env.PORT || 3000;
app.listen(port, function() {
   console.log("To view your app, open this link in your browser: http://localhost:"
+ port);
});
//...
```

Index.js

```
//...
// serve the files out of ./public as our main files
app.use(express.static('public'));
app.set('views', './src/views');
app.set('view engine', 'ejs');
app.use(bodyParser.json());

var title = 'COS Image Gallery Web Application';
// Serve index.ejs
app.get('/', function (req, res) {
  res.render('index', {status: '', title: title});
});

//...
```

Style.css

```
<title><%=title%></title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css
"
     integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u
"
     crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.1.1.min.js"
       integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="
```

```
    crossorigin="anonymous">
</script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
    integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Tx
a"
    crossorigin="anonymous">
</script>

<link rel="stylesheet" href="stylesheets/style.css">
```

Html code

```
<!DOCTYPE html>
<html>

<head>
  <%- include('head-inc'); %>
</head>

<body>
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="/">Home</a></li>
  <li role="presentation"><a href="/gallery">Gallery</a></li>
</ul>
<div class="container">
  <h2>Upload Image to IBM Cloud Object Storage</h2>
  <div class="row">
    <div class="col-md-12">
      <div class="container" style="margin-top: 20px;">
        <div class="row">

          <div class="col-lg-8 col-md-8 well">

            <p class="wellText">Upload your JPG image file here</p>

            <form method="post" enctype="multipart/form-data" action="/">
              <p><input class="wellText" type="file" size="100px" name="img-file" /></p>
              <br/>
              <p><input class="btn btn-danger" type="submit" value="Upload"
/></p>
            </form>

            <br/>
            <span class="notice"><%=status%></span>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</body>

</html>
```

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.**

json

```
{"intents": [

 {

  "intent": "supplieronboarding",

  "confidence": 0.940047025680542

 }

],

"entities": [

{

 "entity": "supplier",

 "value": "supplier"

}

],

"input": {
  "text": " I want to access the supplier on boarding business process"
},
"output": {
 "log_messages": [],
 "text": [
 "To access the Supplier On Boarding business process use the url: "
 ],
 "nodes_visited": [
   "Supplier On Boarding",
   "Provide URL for Supplier on boarding"
 ]
},"context": {
   "conversation_id": "Conversation-ut",
   "system": {
     "dialog_stack": [
     {
       "dialog_node": "root"
     }
     ],
     "dialog_turn_counter": 1,
     "dialog_request_counter": 1
     },
     "url":
"https://vhost001.bpm.ibmcloud.com/bpm/dev/ProcessPortal/dashboards/SYSRP/RESPONSIVE_WORK",
     "action":"click"
  }
}
```

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

```
javascript

if (rep.context.url != undefined) {

if (rep.context.action === "click") {

rep.text=rep.output.text[0] + "<a class=\"btn btn-primary\" href=\""+rep.context.url+"\">Here</a>"

}

}
```