CPS 592 - Cloud Computing & Applications

Instructor: **Dr. Phu Phung**
Department of Computer Science
University of Dayton

Report by, **Abhinava Krishna Sai Reddy Tumukunta** (**101710233**)
tumukuntaa1@udayton.edu, abhinavareddy16@outlook.com

Vinay Varma Hemadri (**101709710**)
hemadriv1@udayton.edu

Project Sprint 1

Part I: **Introduction:**

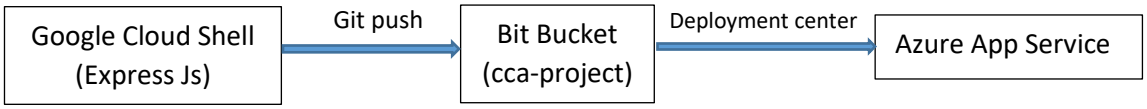
- **Project Objective:**
  To develop Express Web Application based on the Microservices architecture.

- **Project Design:**
  This project features usage of Google Cloud Shell (local machine), BitBucket (repository), Azure App Service (cloud service for one microservice) and Heroku (cloud service for the other microservice).
  Microservice1 is designed to get the year and generate Age and Leap year check.
  Microservice2 is designed to provide the LCM and HCF for any two given input numbers.
  The code for the main Web Application resides in the public BitBucket repository. This web application has 2 microservices each hosted on Azure and Heroku respectively.
  Azure App Service has been linked with a private bitbucket repository to get the code for the microservice1.
  Heroku has been linked with another private bitbucket repository to get the code for the microservice2.

- **Project Development:**

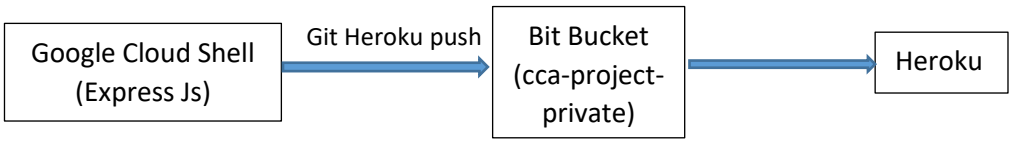
  Microservice1: During the initial phase of the project, to code the microservice1, we have used the express js application from the lab 2. By removing the UI part and used only the call back function to get the messages from server. This code involves displaying the Age and check the Leap year for the year provided in the URL. After successful testing of this application by using web preview from Google Cloud Shell, we pushed this app and related files to the private bitbucket repository – **cca-project** (https://bitbucket.org/cca-hemadriv1-tumukuntaa1/cca-project/src/master/) that has been specifically created to host the code for microservice 1.

Created a new Azure App Service to host this microservice1 application on the internet.
(https://cca-hemadriv1-tumukuntaa1.azurewebsites.net/microservice1?data=2000)
Linked this Azure App Service to our cca-project repository by using the deployment center. This enables the Azure App Service to get the code automatically and deploy directly when ever pushed from the cloud shell to the cca-project bitbucket repository.



Microservice2: In this phase, we have used same express js file from lab2 without UI and added the code to get 2 numbers and output the LCM and GCF of those 2 numbers. The file was then pushed to the private repository specifically created to host the code for this microservice - **cca-project-private** (https://bitbucket.org/cca-hemadriv1-tumukuntaa1/cca-project-private/src/master/).

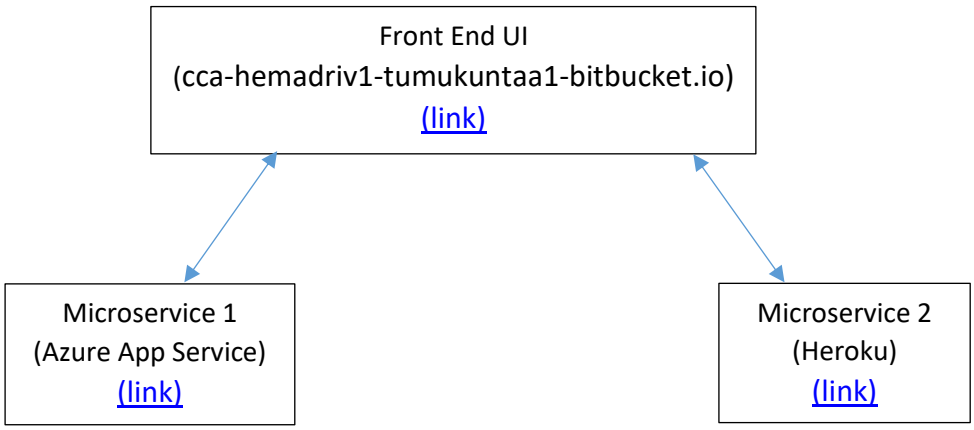
Created new Heroku app service that can host this microservice2 web app on the internet.
(https://cca-team011-microservice2.herokuapp.com/microservice2?a=10&b=20).
Linked this Heroku App hosting service to our bitbucket repository cca-project-private. This enables the direct deployment of our app to the Heroku app service whenever we push our code from our google cloud shell machine to the repository.



Front End UI: In this final phase of our project, the aim is to create an UI based web application that hosts both above microservices in it. For the UI code, we have used the cca-form.html file from the lab2 and renamed it to index.html which makes it easy to load directly in the URL without any additional path names. We have removed the un-necessary codes from this file and provided the code that can invoke microservice1 or microservice2 based on the end user need. This entire has been provided in the codebase section of this report.

We have created a new public repository to host the code for this UI web application – **cca-hemadriv1-tumukuntaa1-bitbucket.io** (https://bitbucket.org/cca-hemadriv1-tumukuntaa1/cca-hemadriv1-tumukuntaa1.bitbucket.io/src/master/).
This repository has been public so that we can directly see our application directly on the internet without using any additional web hosting services by simply going to https:// +our bitbucket repo name (https://cca-hemadriv1-tumukuntaa1.bitbucket.io/). Whenever an end user clicks on the respective sections of microservice1 or microservice and provide the input, in the backend a Web service API on HTTP GET request will call the microservice1 hosted on Azure or microservice2

hosted on Azure respectively and provides the response on the same by using Ajax GET method.



- **Project Achievement:**

Microservice1:

Functional requirements of the microservice 1 have been met by taking 1 input(year) from the user and providing a string with 2 generated fields (Age and Leap year).

Hosted this microservice in the Azure App Service which is directly linked to the private cca-project repository which enables direct deployment by just code push.

Web page (no UI) is responding correctly by providing Age and Leap year.

Microservice2:

Functional requirements of the microservice 2 have been met by taking 2 inputs (values a and b) from the user and providing a string with 2 generated fields (LCM and HCF).

Hosted this microservice in the Heroku which is directly linked to the private cca-project-private repository which enables direct deployment by just code Heroku push.

Web page (no UI) is responding correctly by providing LCM and HCF for the 2 given input values.

Front end UI:

Functional requirements has been met, with one input field and a button Ajax calling the Microservice1 which is hosted on Azure, with the input value and the result from the call will be displayed on the page.

Other 2 input fields and a button Ajax calls the Microservice2 which is hosted on Heroku, with 2 input values and the result from the call will be displayed on the page.

Part II: **Design & Implementation**

Design and implementation of Front End UI: The top class has the code and content that is displayed on the top of our web application which is page description, team member names and email ids. The menu bar has the design and the code related to get current digital time on click. The main class has the design and code related to the Analog clock, microservice1, microservice2 and their Ajax get method codes. In the get method by providing the URL of Azure app services, whenever a user enters a data in microservice 1 filed and hits the button, Ajax get method is called which inturn calls the Azure app service from URL that has been provided in the code and displays the result in the web application as the response. Microservice2 also has the same design with the URL linking it to the Heroku. Please refer Appendix B for the related code.

Implementation and Algorithm of Microservice1: The form code provided below gets the user input of any year. Once the user has clicked the button it will call the microservice1() function.

```
<form action="echo.php" method="GET">
            Enter Any Year: <input name="data" onkeyup="console.log('You have
pressed a key')" id="data">
            <input class="button round" type="button" value="Age and Leap Year"
onclick="microservice1()">
            <div id="response"></div>
         </form>
```

The below code is for microservice1() function.

1. When this function gets called, it will take user input and provides it as suffix to the URL that has been given in the .get() method.
2. This .get() method acts as web service API on HTTP GET request and calls the URL with data.
3. This URL points to Azure App services where our web application is hosted, passes the input data and result is generated.
4. By using the result() function, whatever content displayed on that webpage will be stored in the result and will be displayed as response on our front end UI by using the Ajax Get method.

```
function microservice1(){
                    var input=$("#data").val();
                    if (input.length == 0){return;}
                    $.get("https://cca-hemadriv1-
tumukuntaa1.azurewebsites.net/microservice1?data="+input,
                    function(result){
                        $("#response").html("Response from server:"+result);
                    });
```

```
                $("#data").val("");
            }
```

Implementation and Algorithm of Microservice2: The form code provided below gets the 2 inputs (a and b) from the user. Once the user has clicked the button it will call the microservice2() function.

```
<form action="echo.php" method="GET">
            Enter First Number: <input name="a" onkeyup="console.log('You have
pressed a key')" id="a">
            Enter Second Number: <input name="b" onkeyup="console.log('You have
pressed a key')" id="b">
            <input class="button round" type="button" value="Calculate LCM & HCF"
onclick="microservice2()">
```

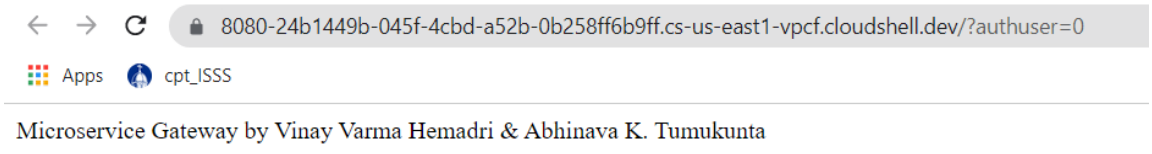
The below code is for microservice2() function.

1. When this function gets called, it will take 2 user inputs a&b and provides it as suffixes to the URL that has been given in the .get() method.
2. This .get() method acts as web service API on HTTP GET request and calls the URL with data.
3. This URL points to Azure App services where our web application is hosted, passes the input data and result is calculated and generated.
4. By using the result() function, whatever content displayed on that webpage will be stored in the result and will be displayed as response on our front end UI by using the Ajax Get method.

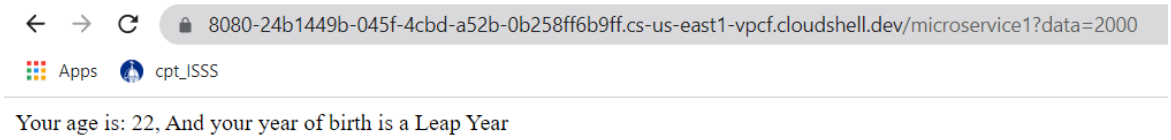
```
function microservice2(){
                    var a=$("#a").val();
                    var b=$("#b").val();
                    $.get("https://cca-team011-
microservice2.herokuapp.com/microservice2?a="+a+"&b="+b,
                    function(result){
                        $("#response1").html("Response from
server:"+result);
                    });
                    $("#a").val("");
                    $("#b").val("");
                }
```

Part III: **Sample Screenshots**

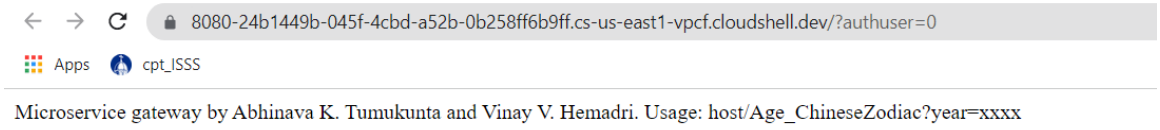
A screenshot of implementing the Microservice1 home is given below,



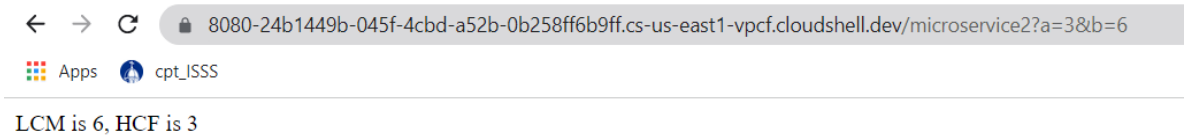
A screenshot of Microservice1 is shown below,



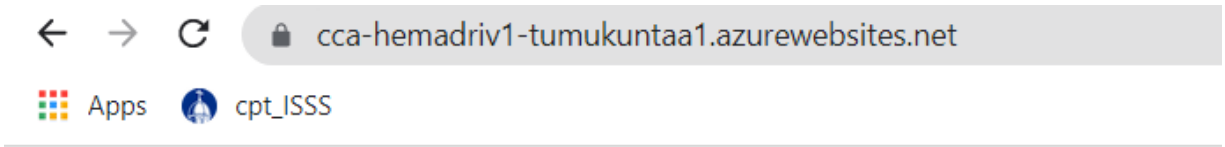
A screenshot of Microservice2 home is attached below,

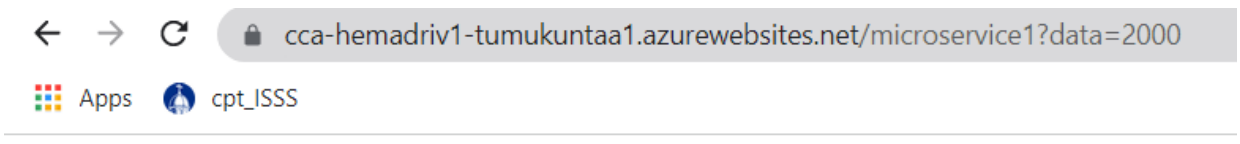


A screenshot of Microservice2 is given below,

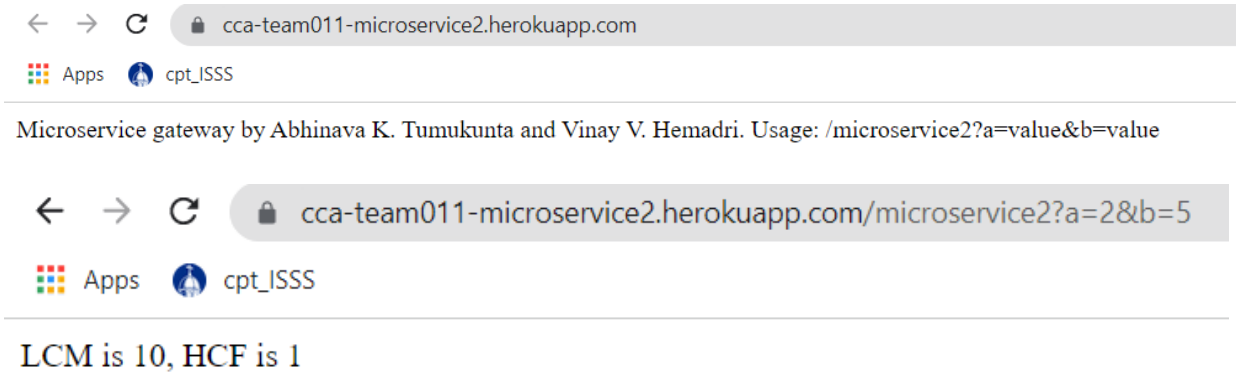


A screenshot of implementing the Azure web app for Microservice1 is shown below,

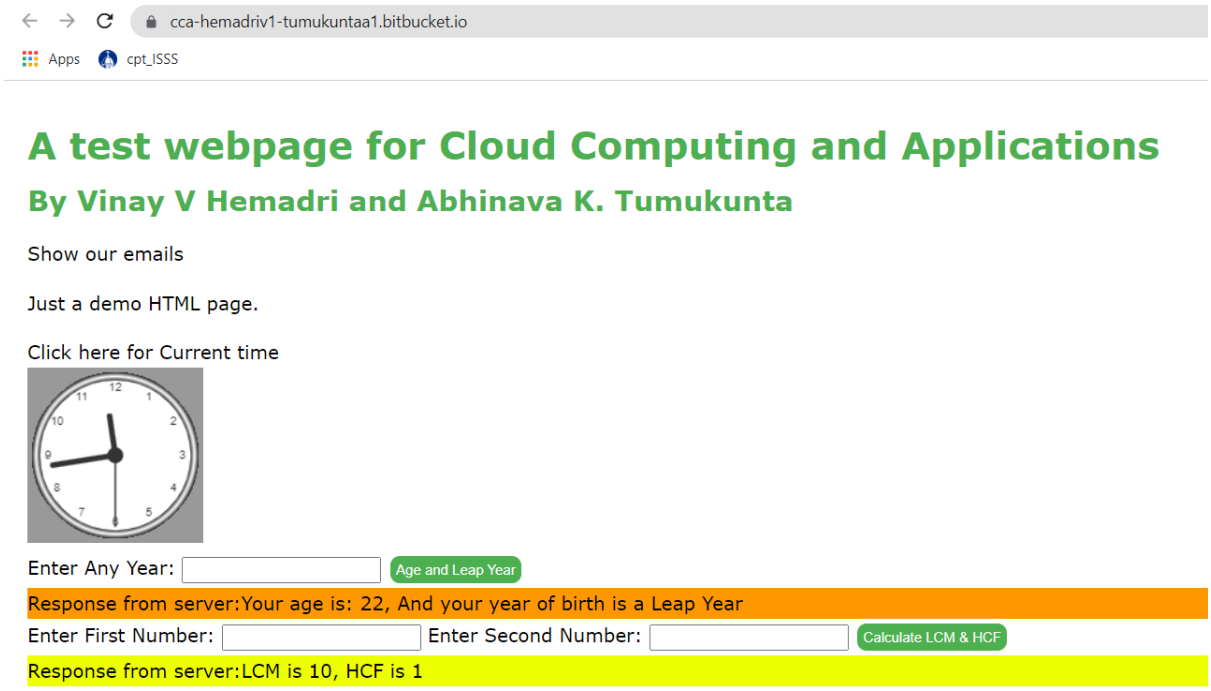




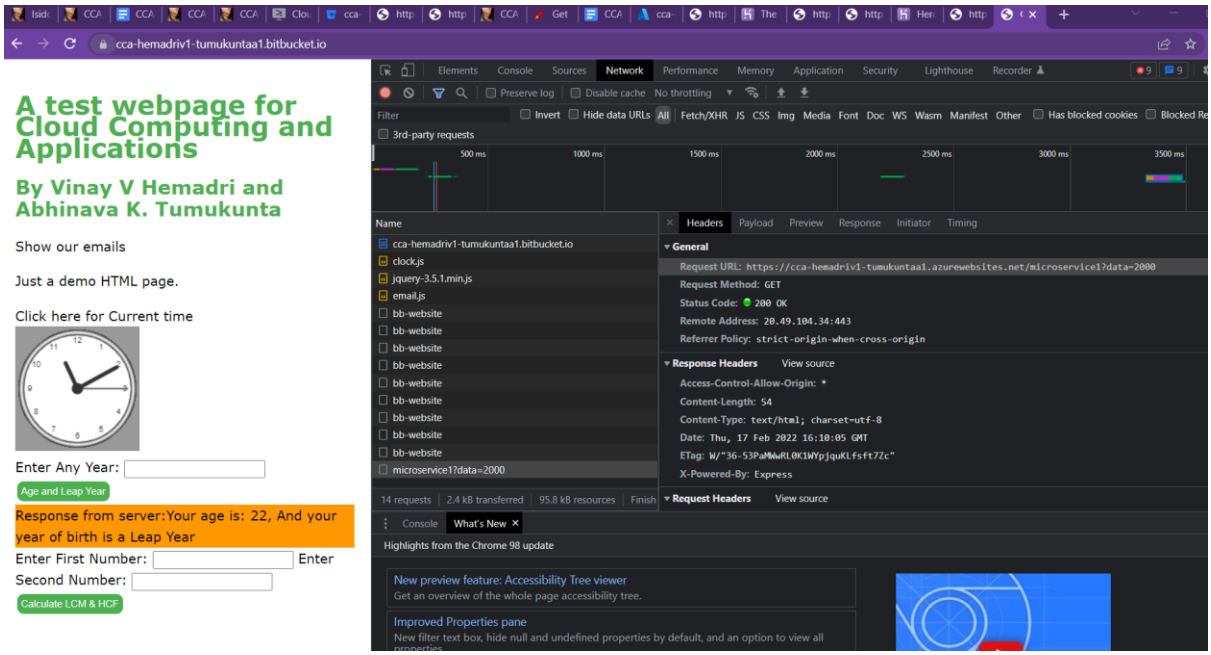
A screenshot of implementing the Heroku app for Microservice2 is shown below,



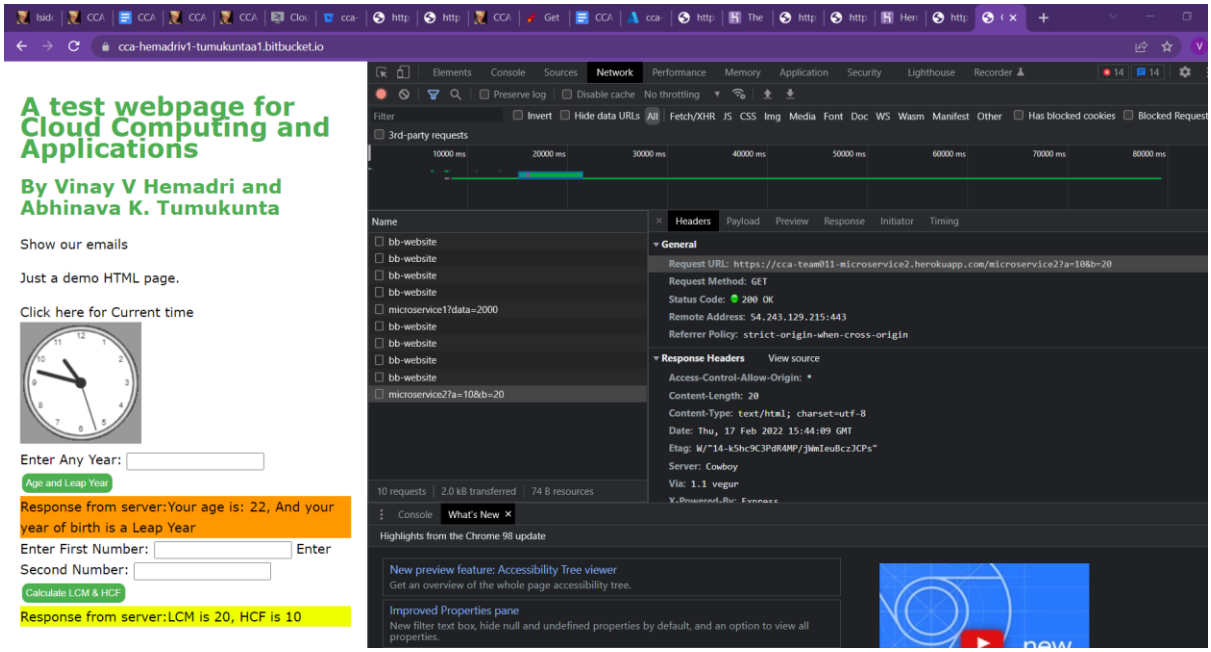
A screenshot of implementing the Project is shown below,



Microservice 1 request url and response header:

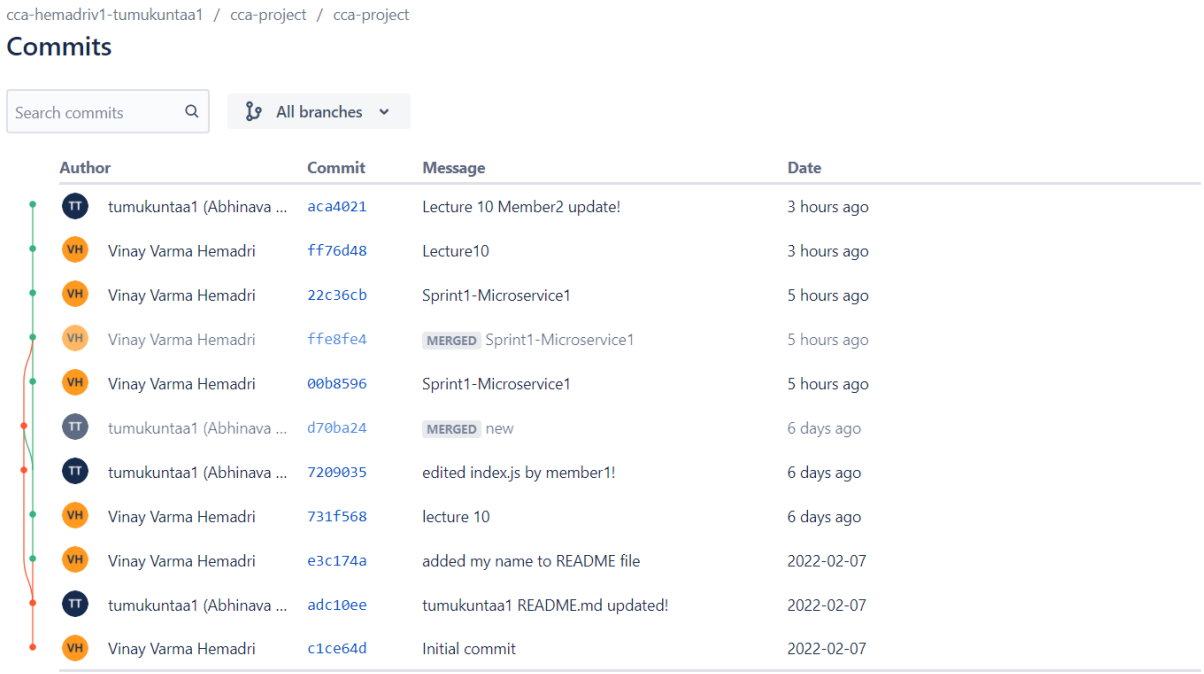


Microservice 2 request url and response header:

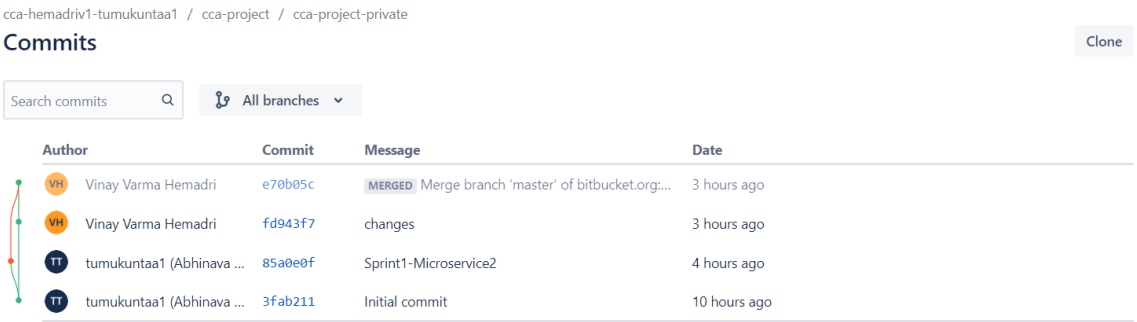


Appendix A – Team Contributions

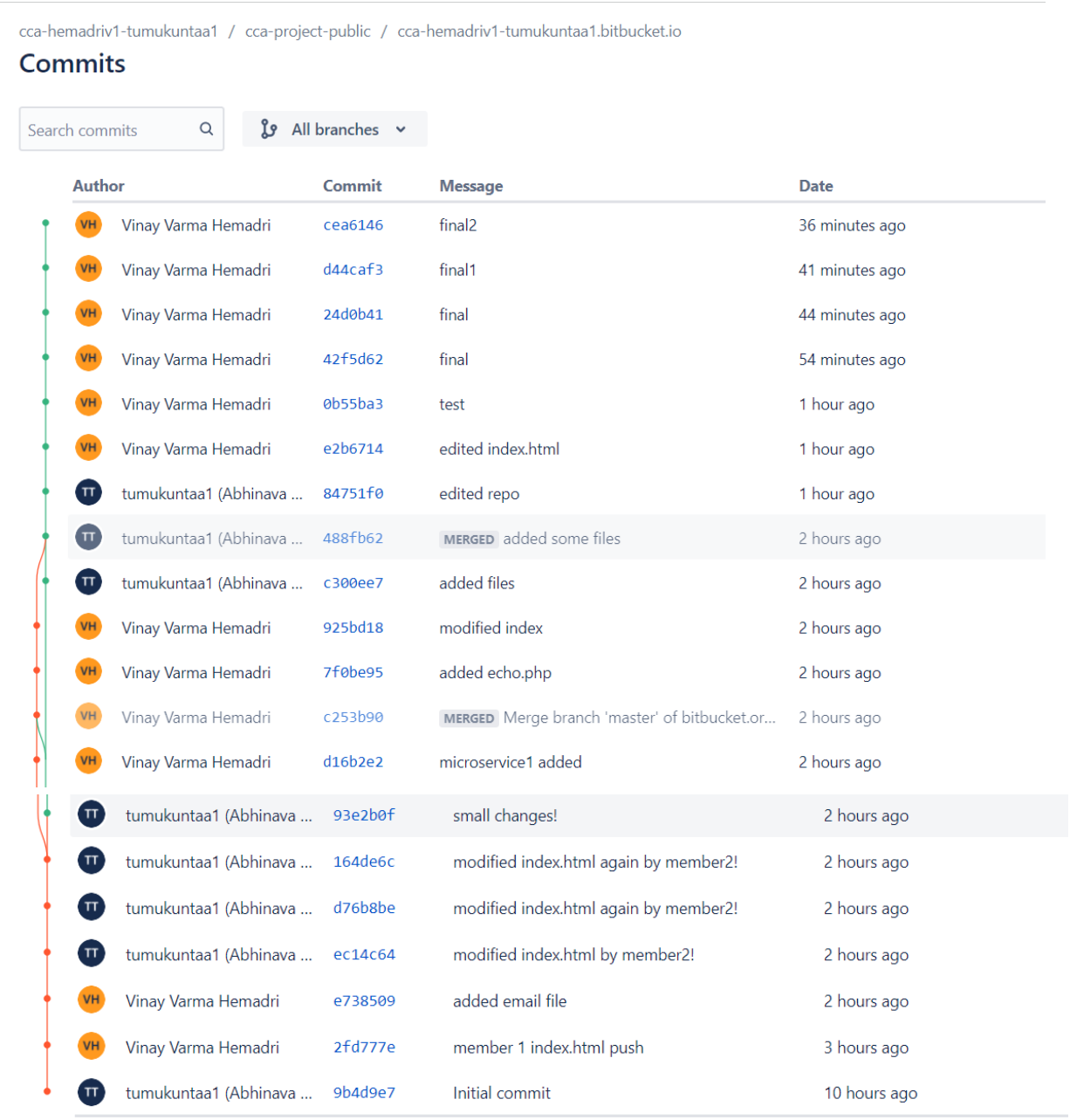
i. Team contributions for private repository ‘cca-project’ used for MicroService1,



ii. Team contributions for private repository ‘cca-project-private’ used for MicroService2,



iii. Team contributions for public repository ‘cca-hemadriv1-tumukuntaa1.bitbucket.io’ used to implement UI,



(PS: In the final hour Team member2-Abhinav has faced error with google cloud shell while working on the public repository of bitbucket.io. So, we have done this part in team member1-Vinay laptop by splitting the work equally. Hence there more commits from vinay in this repository in the final minutes.)

Appendix B – Source Code

i. Front End UI (index.html)

```
<!DOCTYPE html>
<html>
    <head lang="en">
        <meta charset="utf-8">
        <title>CCA-Test page by Team 11</title>
        <link rel="stylesheet" href="https://udayton-
cloud.bitbucket.io/style3.css">
        <script src="https://udayton-cloud.bitbucket.io/clock.js"></script>
        <script src="https://code.jquery.com/jquery-
3.5.1.min.js" crossorigin="anonymous"></script>
        <style>
            .button{
                background-color: #4CAF50;
                border: none;
                color: white;
                padding: 5px;
                text-align: center;
                text-decoration: none;
                display: inline-block;
                font-size: 12px;
                margin: 4px 2px;
                cursor: pointer;
            }
            .round{
                border-radius: 8px;
            }
            #response{
                background-color: #FF9800;
            }
            #response1{
                background-color: #eeff00;
            }
        </style>
    </head>
    <body>
        <div class="container wrapper">
        <div class="top">
        <h1>A test webpage for Cloud Computing and Applications</h1>
        <h2>By Vinay V Hemadri and Abhinava K. Tumukunta</h2>
        <div id="email" onclick="showhideEmail()">Show our emails</div></div>
```

```
        <script src='email.js'></script>
        <div class="wrapper">
        <div class="menubar">
        <p>Just a demo HTML page.</p>
        <div id="mydiv"
                onclick="document.getElementById('mydiv').innerHTML= 'Clicked
time: '+Date()">
            Click here for Current time
        </div>
        <div id="digital-clock"></div></div>
        <div class="main">
        <canvas id="analog-clock" width="150" height="150" style="background-
color: #999"></canvas>
        <script>
            var canvas = document.getElementById("analog-clock");
            var ctx = canvas.getContext("2d");
            var radius=canvas.height /2;
            ctx.translate(radius, radius);
            radius = radius * 0.9;
            setInterval(drawClock, 1000);

            function drawClock(){
                drawFace(ctx, radius);
                drawNumbers(ctx, radius);
                drawTime(ctx, radius);
            }
        </script>
        <form action="echo.php" method="GET">
            Enter Any Year: <input name="data" onkeyup="console.log('You have
pressed a key')" id="data">
            <input class="button round" type="button" value="Age and Leap Year
" onclick="microservice1()">
            <div id="response"></div>
        </form>
        <form action="echo.php" method="GET">
            Enter First Number: <input name="a" onkeyup="console.log('You have
 pressed a key')" id="a">
            Enter Second Number: <input name="b" onkeyup="console.log('You hav
e pressed a key')" id="b">
            <input class="button round" type="button" value="Calculate LCM & H
CF" onclick="microservice2()">
            <script>
                function microservice1(){
                    var input=$("#data").val();
                    if (input.length == 0){return;}
                    $.get("https://cca-hemadriv1-
tumukuntaa1.azurewebsites.net/microservice1?data="+input,
                    function(result){
```

```
                        $("#response").html("Response from server:"+result);
                    });
                    $("#data").val("");
                }
                function microservice2(){
                    var a=$("#a").val();
                    var b=$("#b").val();
                    $.get("https://cca-team011-
microservice2.herokuapp.com/microservice2?a="+a+"&b="+b,
                    function(result){
                        $("#response1").html("Response from server:"+result);
                    });
                    $("#a").val("");
                    $("#b").val("");
                }
            </script>
        </form>
        <div id="response1"></div>

        </div>
        </div>
        </div>
    </body>
</html>
```

ii. MicroService1

```
const express = require('express')
const app = express()
var port = process.env.PORT || 8080;
//app.use(express.static('static'))
app.use(express.urlencoded({extended: false}))
const cors=require('cors')
app.use(cors())
app.listen(port,()=>
    console.log('Express is running on port'+port))
app.get('/',(req,res)=>{
    res.send('Microservice Gateway by Vinay Varma Hemadri & Abhinava K. Tumuku
nta');
})
app.get('/microservice1',function(req,res) {
    var result1 = new Date().getFullYear() - req.query.data
    var result2 = null
    if(req.query.data % 4 == 0){
        result2 = "a Leap Year"
    }
    else{
        result2 = "Not a Leap Year"
    }
    res.send("Your age is: "+result1+", And your year of birth is "+result2)
})
```

iii. MicroService2

```
const express=require('express')
const app=express()
var port= process.env.PORT || 8080;
app.use(express.urlencoded({extended:false}))
const cors=require('cors')
app.use(cors())
app.listen(port,() =>
console.log('Http Server with Express.js is listening on port: ${port}'))

app.get('/',(req,res)=>{
    res.send('Microservice gateway by Abhinava K. Tumukunta and Vinay V. Hemad
ri. Usage: host/Age_ChineseZodiac?year=xxxx')
})
app.get('/Age_ChineseZodiac', function(req, res){
    var result;
    result=new Date().getFullYear() - req.query.data;
    res.send('You are '+result+' years old!')
});
app.get('/microservice1',(req,res)=>{
    res.send("microservice 1 result = "+req.query.data)
})
app.get('/microservice2', function(req, res){
    var a= req.query.a
    var b= req.query.b
    var c= a*b
    while(a != b) {
        if(a>b)
            a=a-b;
        else
            b=b-a;
    }
    lcm=c/a;
    res.send("LCM is "+lcm+", HCF is "+a)
});
```

End of Project Sprint1 Report