Golang Project

**Project Title** : File Downloader

**Name** : Dega Dheepakkh

**SRN** : PES1UG22CS175

**Code :**

**main.go**

package main

import (

  "crypto/tls"

  "fmt"

  "io"

  "net/http"

  "os"

  "sync"

)

func downloadFile(url string, fileName string, wg \*sync.WaitGroup, resultCh chan<- string) {

  defer wg.Done()

  // Create the file to write the downloaded content

  file, err := os.Create(fileName)

  if err != nil {

    resultCh <- fmt.Sprintf("Error creating file %s: %s", fileName, err)

    return

  }

  defer file.Close()

  // Download the file

  client := createHTTPClient()

  resp, err := client.Get(url)

  if err != nil {

    resultCh <- fmt.Sprintf("Error downloading file from %s: %s", url, err)

    return

  }

  defer resp.Body.Close()

  // Write the downloaded content to the file

  \_, err = io.Copy(file, resp.Body)

  if err != nil {

    resultCh <- fmt.Sprintf("Error writing to file %s: %s", fileName, err)

    return

  }

  resultCh <- fmt.Sprintf("File downloaded successfully: %s", fileName)

}

func createHTTPClient() \*http.Client {

  // Disable SSL certificate verification

  tr := &http.Transport{

    TLSClientConfig: &tls.Config{InsecureSkipVerify: true},

  }

  return &http.Client{Transport: tr}

}

func fileDownloadHandler(w http.ResponseWriter, r \*http.Request) {

  url := r.FormValue("url")

  fileName := r.FormValue("fileName")

  var wg sync.WaitGroup

  resultCh := make(chan string)

  wg.Add(1)

  go downloadFile(url, fileName, &wg, resultCh)

  go func() {

    wg.Wait()

    close(resultCh)

  }()

  for result := range resultCh {

    fmt.Fprintln(w, result)

  }

}

func main() {

  http.HandleFunc("/", func(w http.ResponseWriter, r \*http.Request) {

    http.ServeFile(w, r, "index.html")

  })

  http.HandleFunc("/download", fileDownloadHandler)

  fmt.Println("Server started on http://localhost:8080")

  http.ListenAndServe(":8080", nil)

}

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>File Downloader</title>

</head>

<body>

    <h1>File Downloader</h1>

    <form id="downloadForm">

        <label for="url">URL:</label>

        <input type="text" id="url" name="url" required><br><br>

        <label for="fileName">File Name:</label>

        <input type="text" id="fileName" name="fileName" required><br><br>

        <input type="submit" value="Download">

    </form>

    <div id="result"></div>

    <script>

        document.getElementById("downloadForm").addEventListener("submit", function(event) {

            event.preventDefault();

            const formData = new FormData(this);

            const resultDiv = document.getElementById("result");

            // Display "Downloading..." message

            resultDiv.innerText = "Downloading...";

            fetch("/download", {

                method: "POST",

                body: formData

            })

            .then(response => response.text())

            .then(data => {

                // Update resultDiv with the response from the server

                resultDiv.innerText = data;

            })

            .catch(error => {

                // Display error message

                resultDiv.innerText = "Error: " + error.message;

                console.error("Error:", error);

            });

        });

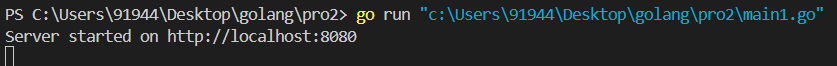
    </script>

</body>

</html>

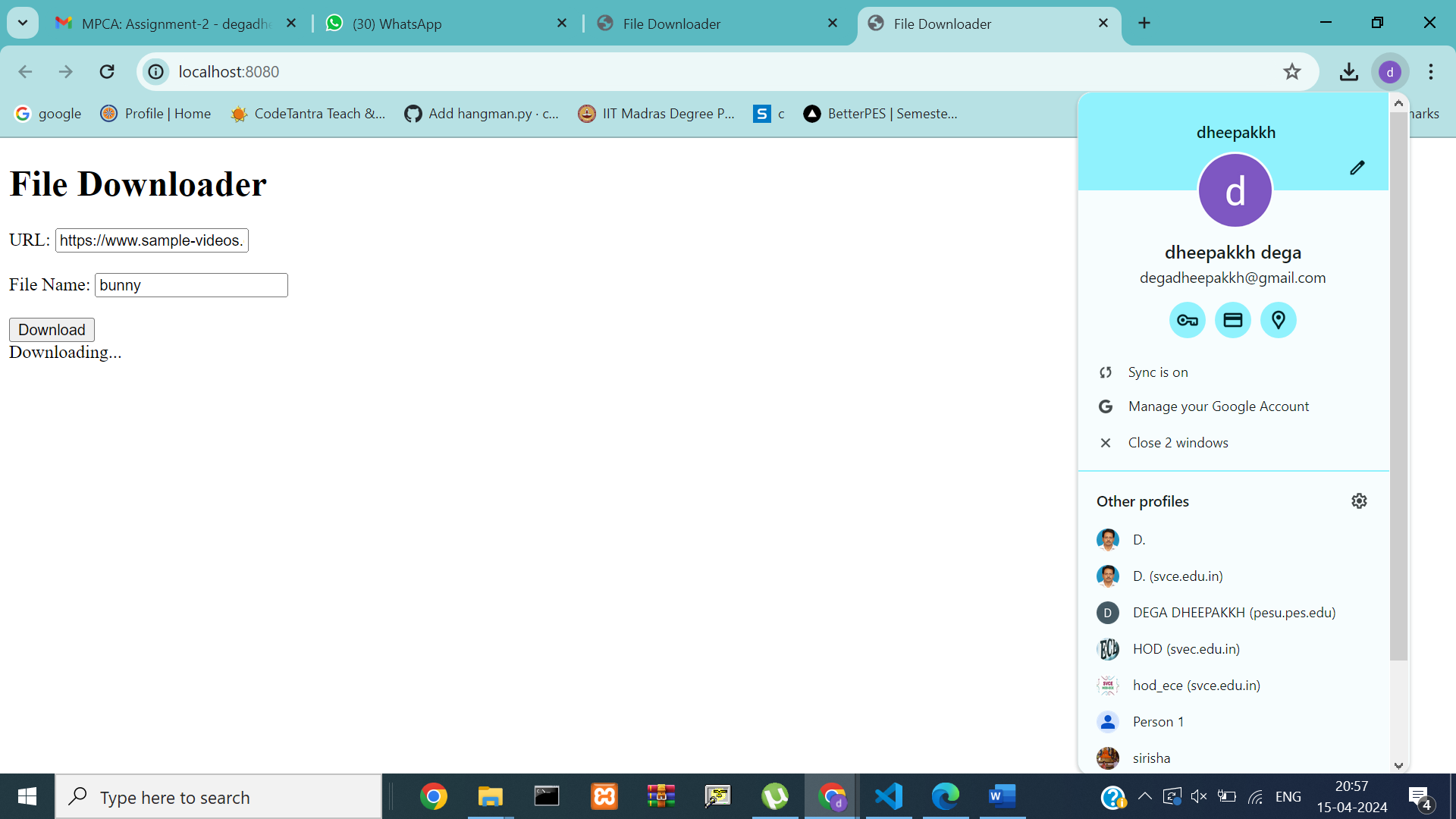
**OUTPUT SCREENSHOTS:**

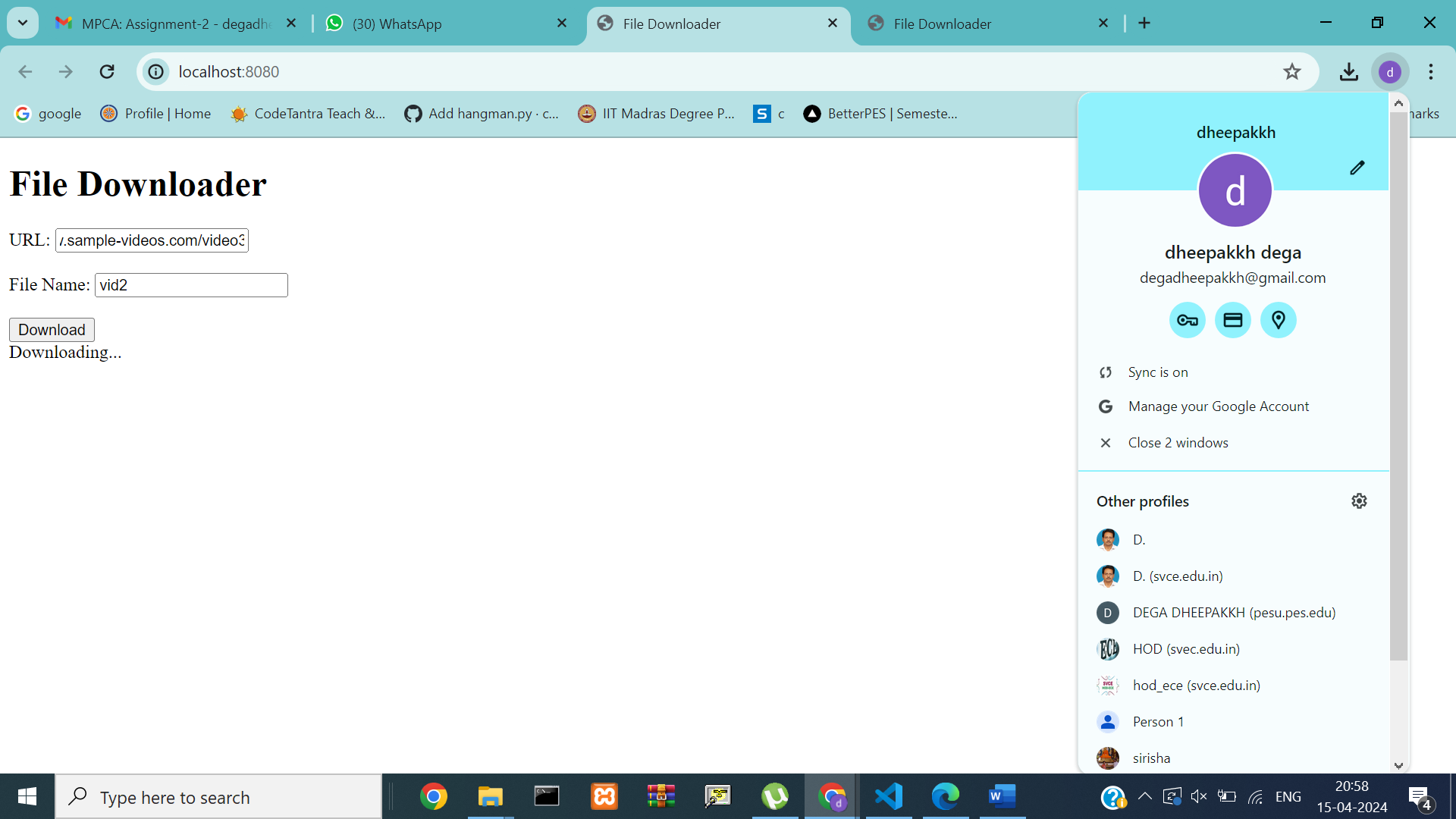
**SERVER:**

****

**CLIENT:**

While downloading:





After downloading:

