

Software Testing

Name: Dev Goel (B20CS090) , Ravi Ramavat (B20CS053)

Project Name: Timely.

- Automated Test Script

Unit Testing Scripts:

a. function to test the time input (hour) by the user.

```
function checkHour(userInputHour, userInputMinute){

    let inputHour = userInputHour;
    let inputMinute = userInputMinute;

    if(inputHour < 0 || inputHour > 24)
    {
        console.log("Error: check the input of hour")
    }
    else if(inputHour == 24 && inputMinute != 0)
    {
        console.log("Error: check the input of hour")
    }
    else
    {
        console.log("No error found.")
    }
}
```

b. function to test the time input (minutes) by the user.

```
function checkMinute(userInputMinute){

    let inputMinute = userInputMinute;
    if(inputMinute < 0 || inputMinute > 60)
    {
        console.log("Error: check the input of minute")
    }
}
```

```

    else
    {
        console.log("No error found.")
    }
}

```

c. function to test the time input (hour) by the user.

```

function checkTheme(userSetTheme){
    let inputTheme = userSetTheme;
    if(inputTheme == 0)
    {
        console.log("White theme set by user.")
    }
    else if(inputTheme == 1)
    {
        console.log("Dark theme set by user.")
    }
    else
    {
        console.log("Error: Invalid theme set attempt by the
user.")
    }
}

```

d. function to test the ignore list of domain.

```

function checkIgnoreListOfDomain(userIgnoreListOfDomain)
{
    const ignoreList = userIgnoreListOfDomain;
    let lengthOfignoreList = ignoreList.length;
    let minLengthPossible = 0;
    let maxLengthPossible = 100;
    if(lengthOfignoreList < minLengthPossible)
    {
        console.log("Error: User tried to add less than
minimum websites possible in ignore list of domain")
    }
    else if(lengthOfignoreList > maxLengthPossible)
    {
        console.log("Error: User tried to add more than
maximum websites possible in ignore list of domain")
    }
}

```

```

    }
    else
    {
        console.log("No error found")
    }
}

```

e. function to check the tabular data.

```

function add(accumulator, a) {
    return accumulator + a;
}

function checkTabularData(userTrackingDataPercentageList){

    let trackingDataPercentageList =
userTrackingDataPercentageList
    for(let i = 0; i < trackingDataPercentageList.length;
i=i+1)
    {
        if(trackingDataPercentageList[i] > 100 ||
trackingDataPercentageList[i] < 0)
        {
            console.log("Error")
        }
    }
    sumOftrackingDataPercentageList =
trackingDataPercentageList.reduce(add, 0)
    if(sumOftrackingDataPercentageList != 100)
    {
        console.log("Error")
    }
    else
    {
        console.log("No error found")
    }
}

```

f. function to test order of tabular data in web analytics table .

```

function testOrderOfTabularData(currentTabularData)

```

```

{
    let error = false;
    for(var i = 0 ; i < currentTabularData.length - 1 ; i
+= 1)
    {
        let object1 = currentTabularData[i]
        let object2 = currentTabularData[i+1]
        if(object1.percentageUsage <
object2.percentageUsage)
        {
            error = true;
        }
    }
    if(error)
    {
        console.log("Error : invalid website sequence in
the table")
    }
    else
    {
        console.log("NO error in order of websites in
tabular data")
    }
}

```

Integration Testing Scripts:

- a. function to check the communication between isDarkThemeON() and user input theme.

```
function checkThemeIntegration(userInput)
{
    checkDarkTheme = isDarkThemeON()
    if(userInput && !checkDarkTheme)
    {
        console.log("Error")
    }
    else if(!userInput && checkDarkTheme)
    {
        console.log("Error")
    }
    else
    {
        console.log("No Error found")
    }
}
```

- b. function to check the integration of restriction list to the web analytics module.

```
async function checkIntegrationRestrictionlist(userInput)
{
    let givenRestrictionList = userInput
    await applyRestrictions(givenRestrictionList);
    appliedRestrictionsList =
applyRestrictions.givenRestrictionList
    if(userInput.length != appliedRestrictionsList)
    {
        console.log("Error in integration of restricted
list");
    }
    else var Error = False
    for(var i = 0 ; i < userInput.length ; i += 1)
    {
        if(userInput[i] != appliedRestrictionsList[i])
```

```

        {
            Error = True;
        }
    }

    if(Error)
    {
        console.log("Error in integration of restricted
list");
    }
    else
    {
        console.log("No Error in integration of restricted
list");
    }
}

```

c. function to check the integration of notification list to the web analytics module.

```

async function checkIntegrationNotificationList(userInput)
{
    let givenNotificationList = userInput
    await setNotifications(givenNotificationList);
    setNotificationList=
setNotifications.givenRestrictionList
    if(userInput.length != setNotificationList)
    {
        console.log("Error in integration of notification
of list");
    }
    else var Error = false
    for(var i = 0 ; i < userInput.length ; i += 1)
    {
        if(userInput[i] != setNotificationList[i])
        {
            Error = true;
        }
    }
}

```

```

        if(Error)
        {
            console.log("Error in integration of notification
list");
        }
        else
        {
            console.log("No Error in integration of
notification list");
        }
    }
}

```

d. function to check integration of notification list with the web analytics module.

```

function checkIntegrationIgnoreList(TabularData)
{
    let ignoreListDomain = getIgnoreListOfDomian()
    let Error = false
    for(var i = 0 ; i < TabularData.length ; i += 1)
    {
        tempWeb = tabularData[i].websiteName;
        if(ignoreListDomain.find(tempWeb))
        {
            Error = true;
        }
    }
    if(Error)
    {
        console.log("Error");
    }
    else
    {
        console.log("No error");
    }
}

```