Project Report

On

MINI APPLICATION WORLD

Submitted by

Syed Fazia R170691
Shaik Sana Sulthana R170741
M.R.Meghana R171078
Soni Kattela R171224

Under the guidance of

Sir A Mahendra

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),

R.K.Valley, Kadapa, Andra Pradesh.

Rajiv Gandhi University of Knowledge Technologies

RK Valley, Kadapa (Dist), Andhra Pradesh, 516330

CERTIFICATE

This is to certify that the project work titled "**MINI APPLICATION WORLD**" submitted by **Syed Fazia** (R170691) , **Shaik Sana Sulthana** (R170741) , **M.R.Meghana** (R171078) , **Kattela Soni** (R171224) in partial fulfillment of the requirements of the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by them under the supervision and guidance.

A Mahendra
Project Internal Guide
Computer Science and Engineering
RGUKT R.K.Valley

P Harinadha
Head of the Department
Computer Science and Engineering
RGUKT R.K.Valley

Submitted for the practical examination held on .....

Internal Examiner **External Examiner**

DECLARATION

We, **Syed Fazia (R170691) , Shaik Sana Sulthana (R170741) , M. R. Meghana (R171078) , Kattela Soni (R171224)** , hereby declare that the project report entitled **" MINI APPLICATION WORLD "** done by us under the guidance of Sir **A Mahendra** is submitted in partial fulfillment for the degree of the Bachelor of Technology in Computer Science and Engineering during the academic session 2021 – 2022 at RGUKT R.K.Valley. We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodies in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

ACKNOWLEDGEMENT

We would to express our sincere gratitude to Sir **A Mahendra,** our project Supervisor for valuable and keen interest throughout the progress of our project. We are grateful to Sir **P.Harinadha**, Head of the Department CSE for providing congenial atmosphere for progressing with our project. We extend our sincere gratitude to the department of Computer Science and Engineering. My sincere thanks to all who have supported me to gain knowledge about actual working involved in various technologies.

Abstract

We are worked on a project entitled Mini Application World. It is just like a hub of day to day used applications. Instead of downloading multiple applications, an app which comprises of multiple applications will make our work quicker and effective.

For now we are worked on integrating all these 4 mini applications and we will be adding more.

They are:

1. Calculator
2. Notes
2. Quiz
3. Dictionary

These are very necessary applications for every student and if all these applications are in one place then it will make our work easier.

1. Calculator:
This free online Calculator can be used for basic computations such as Additions, Subtractions, Multiplications and Divisions.

2. Notes:
Our note taking app helps you capture and prioritize ideas, projects and to-do-lists so nothing falls through the cracks.

3. Quiz
This application is developed for educational purposes allowing the users to prepare the multiple choice questions of Aptitude.

4. Dictionary
A Dictionary app can be a handy tool. It has many featires such as a word of the day to help you to expand your vocabulary.

Technologies and Tools Used:
HTML
CSS
Javascript

We hope our web application eases a student's work.

Contents

1.Introduction

1.1 Description:

Our project entitled Mini Application World, is a hub of day to day used applications. Instead of downloading multiple applications, an app which comprises of multiple applications will make our work quicker and effective. So, we have integrated 4 web applications used by a person in everyday life.

1.2 Purpose:

To ease our work and make it faster and effective. AS this is a web Application, we do not have to worry about storage too.

1.3 Scope:

These are very necessary applications for every student and if all these applications are in one place then it will make our work easier.

1. Calculator:
This free online Calculator can be used for basic computations such as Additions, Subtractions, A and Divisions.

2. Notes:
Our note taking app helps you capture and prioritize ideas, projects and to-do-lists so nothing falls through the cracks.

3. Quiz
This application is developed for educational purposes allowing the users to prepare the multiple choice questions of Aptitude.

4. Dictionary
A Dictionary app can be a handy tool. It has many featires such as a word of the day to help you to expand your vocabulary.

2. Literature Review

2.1 HTML

HTML(Hyper Text Markup Language) is the code that is used o structure a web page and its content.For example,content could be structured withhin a set of paragraphs,a list of bullet points or using images and data tables.As the title sugest this article will give you a basic understanding of html and its functions.HTML is a markup language that defines the structure of your content.HTML consists of series of elements,which you use to enclose or wrap ,different parts of the content to make it appear a certain way.The enclosing tags can make a word or image hyperlink to somewhere else,can italicize words,can make the font bigger or smaller and so on.

2.2 CSS

CSS stands for cascading styles sheets .It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provide an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can be used with any kind of XML documents including plain XML ,SVG and XUL. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web application and user interfaces for many mobile applications.

2.3 Javascript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

3.Software Requirement Specification

Functional Requirements:

Functinal requirements show the operation and activities the system must be able to perform.The functional Requirements of Users.

- Secure Access
- It allows Multiple user access at a time.
- It allows to use and compute basic operations.
- It allows to NoteMaking the Important information.
- It allows to Search the meanings of the Unkown Words.
- It Allows to practise the Objective Questions .

Non-functional Requirements:

Usability:

The system is designed with completely automated process hence there is no or less user intervention

.

Realiability:

The System is more reliable because of the qualities that are inherited from the chosen platform html,css,Javascript.

Performence:

This System is developing in the high level languages and using the advanced fron-end and back-end technologies it will give response to the end user on client System with in very less time.

Supportability:

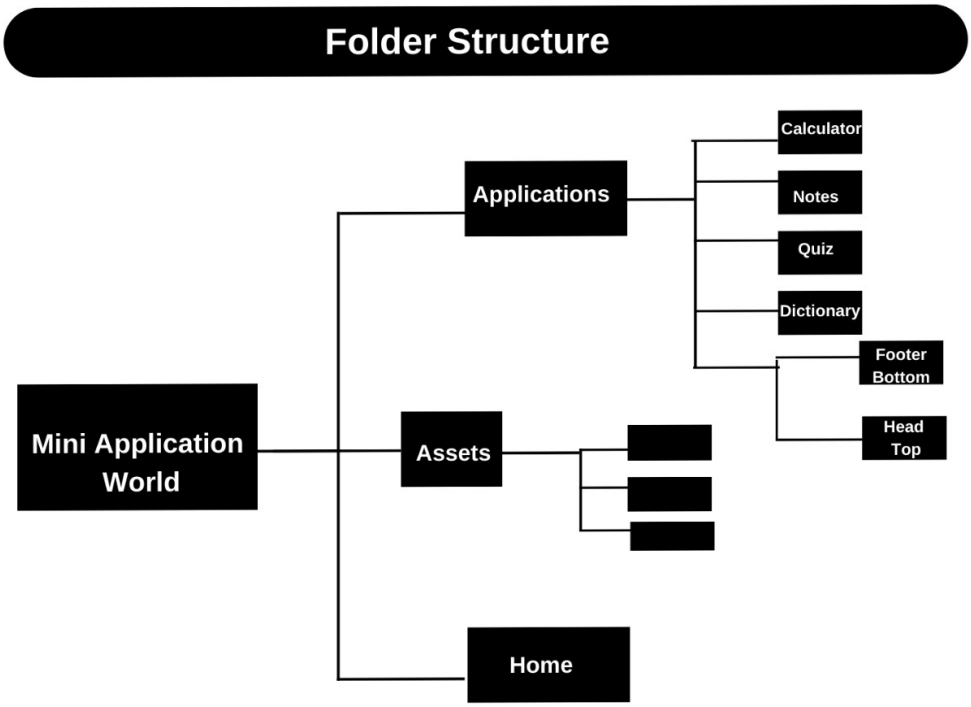
The System is desingned to be the cross platform supportable .The system is Supported on wide range of hardware and any softaware platform.

Scalability:

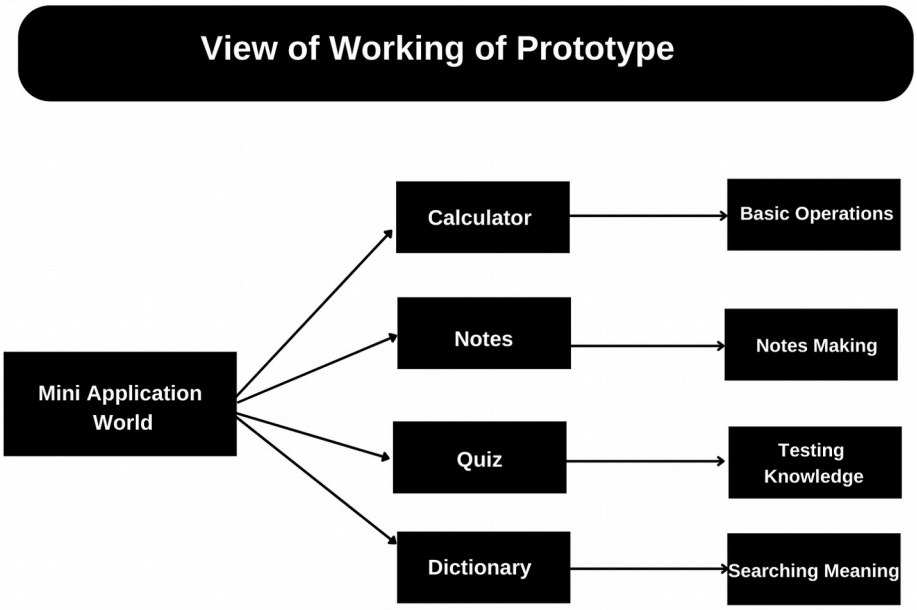
The System is designed to be test the scalability by adding additional load to the website.

4. System Design

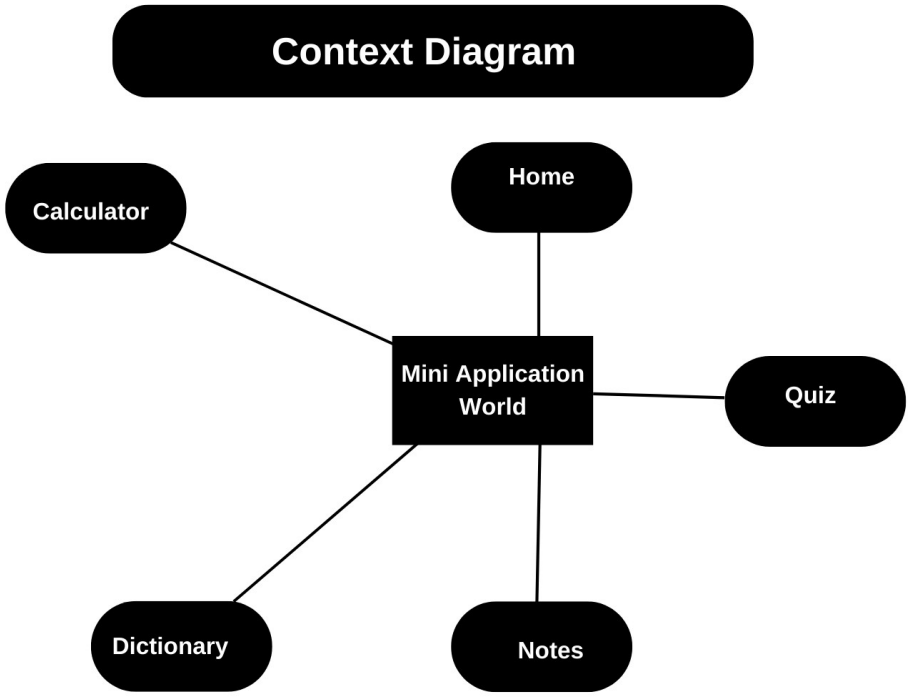
4.1 Folder Structure



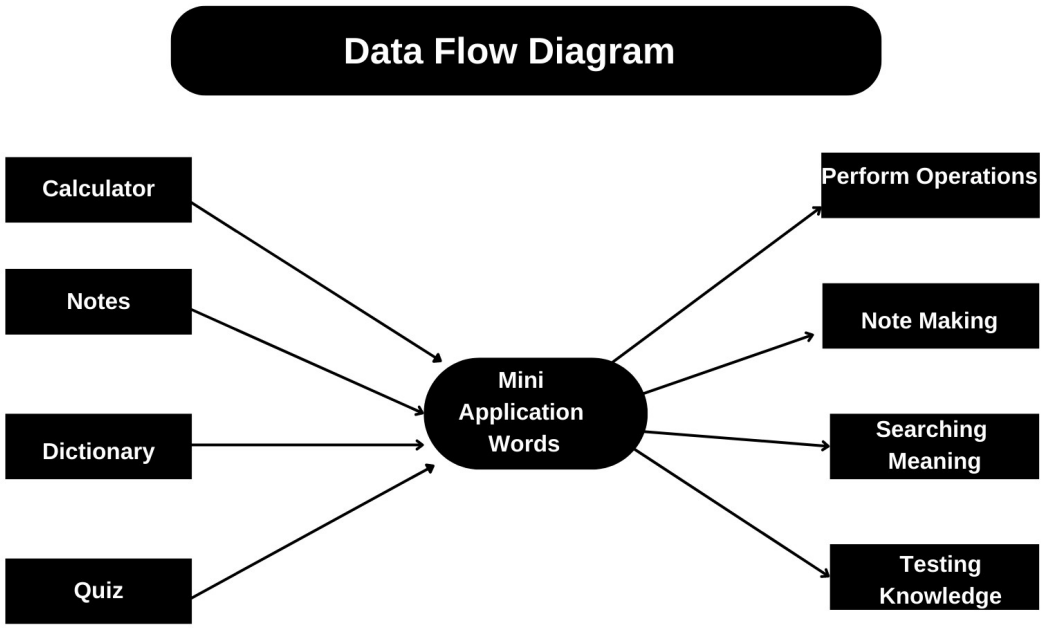
4.2 Pictoral View of Prototype



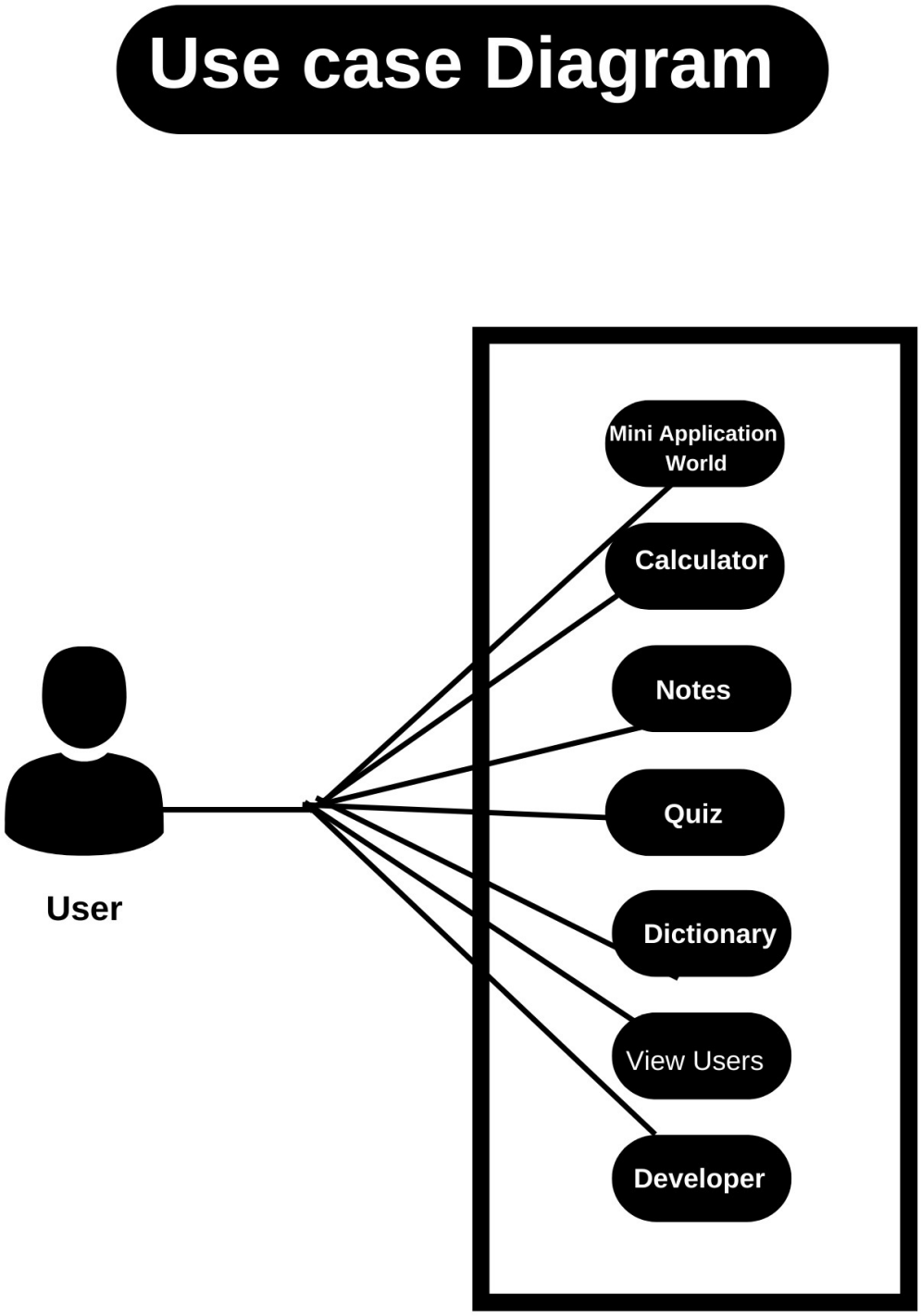
4.3 Context Diagram



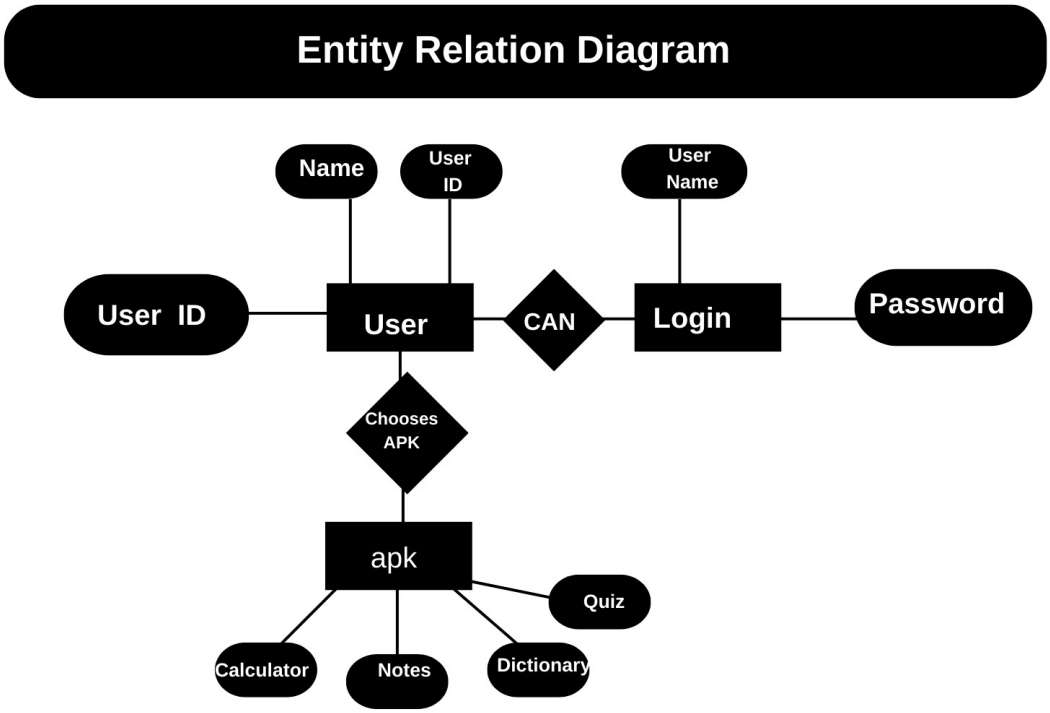
4.3 Data Flow Diagram



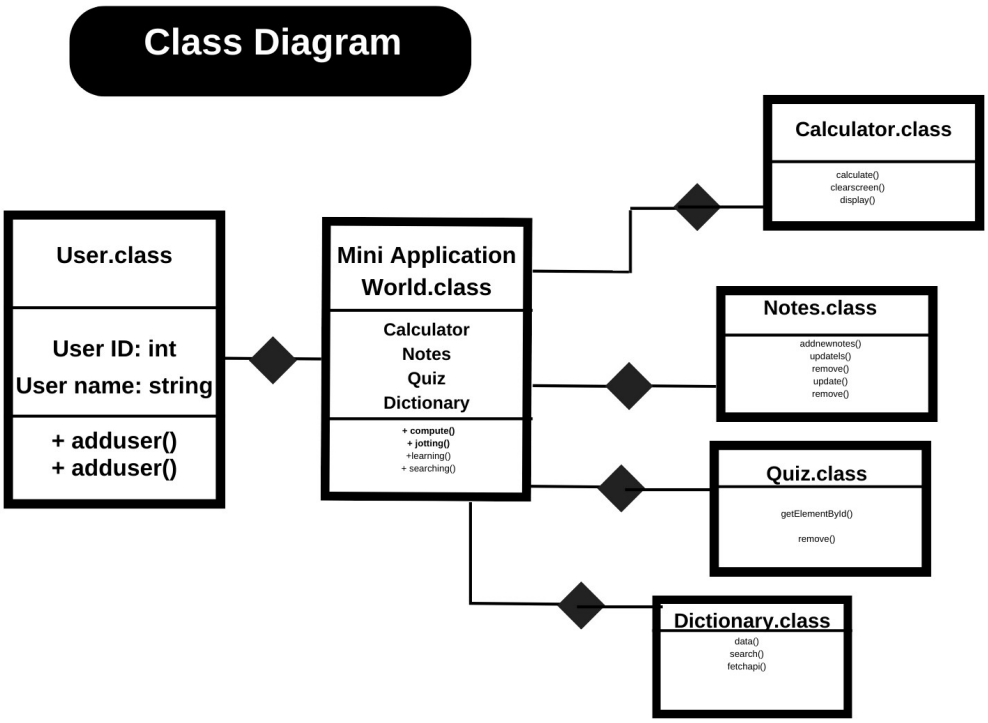
4.5 Use Case Diagram



4.6 ER Diagram



4.7 Class Diagram



5.Coding Or Implementation

Implementation is the stage of the project when the threoretical design is turned out into work system.Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user ,confidence that the new system will work and be affective

The implementation stage involves careful planning,investigation of the existing system and it's constrains on implementation,designing of methods to achieve changeover and evaluation of changeover methods

Home.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Web Application World </title>
      <link rel="stylesheet" href="Home.css"/>
   </head>

   <body>

            <marquee scrollamount="6" repeat="2"><font face="impact" size="300px"
color="white">Mini Web Application World</font></marquee>
       <table align="center">
       <tr>
       <td>
        <a href="Calculator/index.html"> <img src="calculator.jpg" width="250"
height="300" title="Calculator"> </a>
       </td>
       <td>

       </td>
       <td>

       </td>
       <td>

       </td>
       <td>

       </td>
       <td>
```

```
<a href="Notes/index.html"> <img src="Notes.png" width="250" height="300"
title="Notes"> </a>
        </td>
        <td>

        </td>
        <td>

        </td>
        <td>

        </td>
        <td>

        </td>
        <td>

        </td>
        <td>
        <a href="Quiz/index.html" > <img src="Quiz.png" width="250" height="300"
title="Quiz"> </a>
        </td>
        <td>

        </td>
        <td>

        </td>
        <td>

        </td>
        <td>

        </td>
        <td>

        </td>
        <td>
        <a href="Dictionary/index.html" > <img src="dict.png" width="250"
height="300" title="Dictionary"> </a>
        </td>
        </tr>
        </table>
   </body>
</html>
```

Home.css

```
body{
   background: linear-gradient(-45deg,#ee7752,#e73c7e,#23a6d5,#23d5ab);
        height: 100vh;
}
body,a{
   align-items: center;
}
h1{
   text-align: center;
}
table,td{

   border-collapse:collapse;

}
```

Calculator/index.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">

<head>
  <meta charset="utf-8">
  <title>Simple Calculator using HTML, CSS and JavaScript</title>
  <link rel="stylesheet" href="styles.css">
</head>

<body>

<table class="calculator" >
  <tr>
    <td colspan="3"> <input class="display-box" type="text" id="result" disabled />
</td>

    <!-- clearScreen() function clears all the values -->
    <td> <input type="button" value="C" onclick="clearScreen()" id="btn" /> </td>
  </tr>
  <tr>
```

```
    <!-- display() function displays the value of clicked button -->
    <td> <input type="button" value="1" onclick="display('1')" /> </td>
    <td> <input type="button" value="2" onclick="display('2')" /> </td>
    <td> <input type="button" value="3" onclick="display('3')" /> </td>
    <td> <input type="button" value="/" onclick="display('/')" /> </td>
  </tr>
  <tr>
    <td> <input type="button" value="4" onclick="display('4')" /> </td>
    <td> <input type="button" value="5" onclick="display('5')" /> </td>
    <td> <input type="button" value="6" onclick="display('6')" /> </td>
    <td> <input type="button" value="-" onclick="display('-')" /> </td>
  </tr>
  <tr>
    <td> <input type="button" value="7" onclick="display('7')" /> </td>
    <td> <input type="button" value="8" onclick="display('8')" /> </td>
    <td> <input type="button" value="9" onclick="display('9')" /> </td>
    <td> <input type="button" value="+" onclick="display('+')" /> </td>
  </tr>
  <tr>
    <td> <input type="button" value="." onclick="display('.')" /> </td>
    <td> <input type="button" value="0" onclick="display('0')" /> </td>

    <!-- calculate() function evaluates the mathematical expression -->
    <td> <input type="button" value="=" onclick="calculate()" id="btn" /> </td>
    <td> <input type="button" value="*" onclick="display('*')" /> </td>
  </tr>
</table>

<script type="text/javascript" src="script.js"></script>

</body>

</html>
```

Calculator/styles.css

```
@import url('https://fonts.googleapis.com/css2?family=Orbitron&display=swap');
.calculator {
    padding: 10px;
    border-radius: 1em;
    height: 380px;
    width: 400px;
    margin: auto;
    background-color: #191b28;
    box-shadow: rgba(0, 0, 0, 0.19) 0px 10px 20px, rgba(0, 0, 0, 0.23) 0px 6px 6px;
}
body{
    background-color: aqua;
}
.display-box {
    font-family: 'Orbitron', sans-serif;
    background-color: #dcdbe1;
    border: solid black 0.5px;
    color: black;
    border-radius: 5px;
    width: 100%;
    height: 65%;
}
#btn {
    background-color: #fb0066;
}
input[type=button] {
    font-family: 'Orbitron', sans-serif;
    background-color: #64278f;
    color: white;
    border: solid black 0.5px;
   width: 100%;
    border-radius: 5px;
    height: 70%;
    outline: none;
}
input:active[type=button] {
    background: #e5e5e5;
    -webkit-box-shadow: inset 0px 0px 5px #c1c1c1;
    -moz-box-shadow: inset 0px 0px 5px #c1c1c1;
    box-shadow: inset 0px 0px 5px #c1c1c1;
}
```

Calculator/script.js

```
// This function clear all the values
function clearScreen() {
    document.getElementById("result").value = "";
}

// This function display values
function display(value) {
    document.getElementById("result").value += value;
}

// This function evaluates the expression and returns result
function calculate() {
    var p = document.getElementById("result").value;
    var q = eval(p);
    document.getElementById("result").value = q;
}
```

Notes/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/
css/all.min.css" integrity="sha512-
1PKOgIY59xJ8Co8+NE6FZ+LOAZKjy+KY8iq0G4B3CyeY6wYHN3yt9PW0XpSriVlkM
Xe40PTKnXrLnZ9+fkDaog==" crossorigin="anonymous" />
    <link rel="stylesheet" href="style.css" />
    <title>Notes App</title>
  </head>
  <body>
    <button class="add" id="add">
      <i class="fas fa-plus"></i> Add note
    </button>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/marked/1.2.2/marked.min.js"></
script>
    <script src="script.js"></script>
  </body>
</html>
```

Notes/styles.css

```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:wght@200;400&display=swap');

* {
  box-sizing: border-box;
  outline: none;
}

body {
  background-color: #7bdaf3;
  font-family: 'Poppins', sans-serif;
  display: flex;
  flex-wrap: wrap;
  margin: 0;
  padding-top: 3rem;
}

.add {
  position: fixed;
  top: 1rem;
  right: 1rem;
  background-color: #9ec862;
  color: #fff;
  border: none;
  border-radius: 3px;
  padding: 0.5rem 1rem;
  cursor: pointer;
}

.add:active {
  transform: scale(0.98);
}

.note {
  background-color: #fff;
  box-shadow: 0 0 10px 4px rgba(0, 0, 0, 0.1);
  margin: 30px 20px;
  height: 400px;
  width: 400px;
  overflow-y: scroll;
}
```

```
.note .tools {
  background-color: #9ec862;
  display: flex;
  justify-content: flex-end;
  padding: 0.5rem;
}

.note .tools button {
  background-color: transparent;
  border: none;
  color: #fff;
  cursor: pointer;
  font-size: 1rem;
  margin-left: 0.5rem;
}

.note textarea {
  outline: none;
  font-family: inherit;
  font-size: 1.2rem;
  border: none;
  height: 400px;
  width: 100%;
  padding: 20px;
}

.main {
  padding: 20px;
}

.hidden {
  display: none;
}
```

Notes/script.js

```
const addBtn = document.getElementById('add')

const notes = JSON.parse(localStorage.getItem('notes'))

if(notes) {
    notes.forEach(note => addNewNote(note))
}

addBtn.addEventListener('click', () => addNewNote())
```

```
function addNewNote(text = '') {
    const note = document.createElement('div')
    note.classList.add('note')

    note.innerHTML = `
    <div class="tools">
        <button class="edit"><i class="fas fa-edit"></i></button>
        <button class="delete"><i class="fas fa-trash-alt"></i></button>
    </div>
    <div class="main ${text ? "" : "hidden"}"></div>
    <textarea class="${text ? "hidden" : ""}"></textarea>
    `

    const editBtn = note.querySelector('.edit')
    const deleteBtn = note.querySelector('.delete')
    const main = note.querySelector('.main')
    const textArea = note.querySelector('textarea')

    textArea.value = text
    main.innerHTML = marked(text)

    deleteBtn.addEventListener('click', () => {
        note.remove()

        updateLS()
    })

    editBtn.addEventListener('click', () => {
        main.classList.toggle('hidden')
        textArea.classList.toggle('hidden')
    })

    textArea.addEventListener('input', (e) => {
        const { value } = e.target

        main.innerHTML = marked(value)

        updateLS()
    })

    document.body.appendChild(note)
}
```

```
function updateLS() {
   const notesText = document.querySelectorAll('textarea')

   const notes = []

   notesText.forEach(note => notes.push(note.value))

   localStorage.setItem('notes', JSON.stringify(notes))
}
```

Quiz/home.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Quick Quiz</title>
    <link rel="stylesheet" href="app.css" />
  </head>
  <body>
    <div class="container">
      <div id="home" class="flex-center flex-column">
        <h1>Quick Quiz</h1>
        <a class="btn" href="index.html">Play</a>
        <a class="btn" href="highscores.html">High Scores</a>
      </div>
    </div>
  </body>
</html>
```

Quiz/app.css

```
   :root {
  background-color: #ecf5ff;
  font-size: 62.5%;
}
 *{
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
  margin: 0;
  padding: 0;
  Color: #333;
}
```

```
h1,
h2,
h3,
h4 {
  margin-bottom: 1rem;
}

h1 {
  font-size: 5.4rem;
  color: #56a5eb;
  margin-bottom: 5rem;
}

h1 > span {
  font-size: 2.4rem;
  font-weight: 500;
}

h2 {
  font-size: 4.2rem;
  margin-bottom: 4rem;
  font-weight: 700;
}

h3 {
  font-size: 2.8rem;
  font-weight: 500;
}

/* UTILITIES */

.container {
  width: 100vw;
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  max-width: 80rem;
  margin: 0 auto;
  padding: 2rem;
}

.container > * {
  width: 100%;
}
```

```
.flex-column {
  display: flex;
  flex-direction: column;
}

.flex-center {
  justify-content: center;
  align-items: center;
}

.justify-center {
  justify-content: center;
}

.text-center {
  text-align: center;
}

.hidden {
  display: none;
}

/* BUTTONS */
.btn {
  font-size: 1.8rem;
  padding: 1rem 0;
  width: 20rem;
  text-align: center;
  border: 0.1rem solid #56a5eb;
  margin-bottom: 1rem;
  text-decoration: none;
  color: #56a5eb;
  background-color: white;
}

.btn:hover {
  cursor: pointer;
  box-shadow: 0 0.4rem 1.4rem 0 rgba(86, 185, 235, 0.5);
  transform: translateY(-0.1rem);
  transition: transform 150ms;
}

.btn[disabled]:hover {
  cursor: not-allowed;
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Quick Quiz - Play</title>
    <link rel="stylesheet" href="app.css" />
    <link rel="stylesheet" href="game.css" />
  </head>
  <body>
    <div class="container">
      <div id="loader"></div>
      <div id="game" class="justify-center flex-column hidden">
        <div id="hud">
          <div id="hud-item">
            <p id="progressText" class="hud-prefix">
              Question
            </p>
            <div id="progressBar">
              <div id="progressBarFull"></div>
            </div>
          </div>
          <div id="hud-item">
            <p class="hud-prefix">
              Score
            </p>
            <h1 class="hud-main-text" id="score">
              0
            </h1>
          </div>
        </div>
        <h2 id="question"></h2>
        <div class="choice-container">
          <p class="choice-prefix">A</p>
          <p class="choice-text" data-number="1"></p>
        </div>
        <div class="choice-container">
          <p class="choice-prefix">B</p>
          <p class="choice-text" data-number="2"></p>
        </div>
        <div class="choice-container">
          <p class="choice-prefix">C</p>
          <p class="choice-text" data-number="3"></p>
```

```
.btn[disabled]:hover {
  cursor: not-allowed;
  box-shadow: none;
  transform: none;
}

/* FORMS */
form {
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
}

input {
  margin-bottom: 1rem;
  width: 20rem;
  padding: 1.5rem;
  font-size: 1.8rem;
  border: none;
  box-shadow: 0 0.1rem 1.4rem 0 rgba(86, 185, 235, 0.5);
}

input::placeholder {
  color: #aaa;
}
```

Quiz/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Quick Quiz</title>
    <link rel="stylesheet" href="app.css" />
  </head>
  <body>
    <div class="container">
      <div id="home" class="flex-center flex-column">
        <h1>Quick Quiz</h1>
        <a class="btn" href="index.html">Play</a>
        <a class="btn" href="highscores.html">High Scores</a>
      </div>  </div>
  </body>
</html>
```

game1.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Quick Quiz - Play</title>
    <link rel="stylesheet" href="app.css" />
    <link rel="stylesheet" href="game.css" />
  </head>
  <body>
    <div class="container">
      <div id="loader"></div>
      <div id="game" class="justify-center flex-column hidden">
        <div id="hud">
          <div id="hud-item">
            <p id="progressText" class="hud-prefix">
              Question
            </p>
            <div id="progressBar">
              <div id="progressBarFull"></div>
            </div>
          </div>
          <div id="hud-item">
            <p class="hud-prefix">
              Score
            </p>
            <h1 class="hud-main-text" id="score">
              0
            </h1>
          </div>
        </div>
        <h2 id="question"></h2>
        <div class="choice-container">
          <p class="choice-prefix">A</p>
          <p class="choice-text" data-number="1"></p>
        </div>
        <div class="choice-container">
          <p class="choice-prefix">B</p>
          <p class="choice-text" data-number="2"></p>
        </div>
        <div class="choice-container">
          <p class="choice-prefix">C</p>
```

game1.html

```
  <p class="choice-text" data-number="3"></p>
        </div>
        <div class="choice-container">
          <p class="choice-prefix">D</p>
          <p class="choice-text" data-number="4"></p>
        </div>
      </div>
    </div>
    <script src="game1.js"></script>
  </body>
</html>
```

game1.js

```
 const question = document.getElementById('question');
 const choices = Array.from(document.getElementsByClassName('choice-text'));
 const progressText = document.getElementById('progressText');
 const scoreText = document.getElementById('score');
 const progressBarFull = document.getElementById('progressBarFull');
 const loader = document.getElementById('loader');
 const game = document.getElementById('game');
 let currentQuestion = {};
 let acceptingAnswers = false;
 let score = 0;
 let questionCounter = 0;
 let availableQuesions = [];
 let questions = [];
 fetch(
    'https://opentdb.com/api.php?
 amount=10&category=9&difficulty=easy&type=multiple'
 )
    .then((res) => {
       return res.json();
    })
    .then((loadedQuestions) => {
       questions = loadedQuestions.results.map((loadedQuestion) => {
          const formattedQuestion = {
             question: loadedQuestion.question,
          };

          const answerChoices = [...loadedQuestion.incorrect_answers];
          formattedQuestion.answer = Math.floor(Math.random() * 4) + 1;
          answerChoices.splice(
             formattedQuestion.answer – 1,0,loadedQuestion.correct_answer
          );
```

Dictionary/index.html

```
<!DOCTYPE html>

<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Dictionary App in JavaScript | CodingNepal</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- CDN Link for Icons -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?
family=Material+Icons">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
  </head>
        <!--<marque scrollamount='80'><font face="impact"size="4" color="red">
SEARCH FOR MEANING</font></marque>-->
  <body>
    <div class="wrapper">
      <header>English Dictionary</header>
      <div class="search">
        <input type="text" placeholder="Search a word" required spellcheck="false">
        <i class="fas fa-search"></i>
        <span class="material-icons">close</span>
      </div>
      <p class="info-text">Type any existing word and press enter to get meaning,
example, synonyms, etc.</p>
      <ul>
        <li class="word">
          <div class="details">
            <p>__</p>
            <span>_ _</span>
          </div>
          <i class="fas fa-volume-up"></i>
        </li>
        <div class="content">
          <li class="meaning">
            <div class="details">
              <p>Meaning</p>
              <span>___</span>
            </div>
          </li>
          <li class="example">
```

```
 <div class="details">
            <p>Example</p>
            <span>___</span>
          </div>
        </li>
        <li class="synonyms">
          <div class="details">
            <p>Synonyms</p>
            <div class="list"></div>
          </div>
        </li>
      </div>
    </ul>
  </div>

  <script src="script.js"></script>

 </body>
</html>
```

Dictionary/styles.css

```
/* Import Google Font - Poppins */
@import url('https://fonts.googleapis.com/css2?
family=Poppins:wght@400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  background: #4D59FB;
}
::selection{
  color: #fff;
  background: #4D59FB;
}
.wrapper{
  width: 420px;
```

```
 border-radius: 7px;
  background: #fff;
  padding: 25px 28px 45px;
  box-shadow: 7px 7px 20px rgba(0, 0, 0, 0.05);
}
.wrapper header{
  font-size: 28px;
  font-weight: 500;
  text-align: center;
}
.wrapper .search{
  position: relative;
  margin: 35px 0 18px;
}
.search input{
  height: 53px;
  width: 100%;
  outline: none;
  font-size: 16px;
  border-radius: 5px;
  padding: 0 42px;
  border: 1px solid #999;
}
.search input:focus{
  padding: 0 41px;
  border: 2px solid #4D59FB;
}
.search input::placeholder{
  color: #B8B8B8;
}
.search :where(i, span){
  position: absolute;
  top: 50%;
  color: #999;
  transform: translateY(-50%);
}
.search i{
  left: 18px;
  pointer-events: none;
  font-size: 16px;
}
.search input:focus ~ i{
  color: #4D59FB;
}
.search span{
```

```
 right: 15px;
  cursor: pointer;
  font-size: 18px;
  display: none;
}
.search input:valid ~ span{
  display: block;
}
.wrapper .info-text{
  font-size: 13px;
  color: #9A9A9A;
  margin: -3px 0 -10px;
}
.wrapper.active .info-text{
  display: none;
}
.info-text span{
  font-weight: 500;
}
.wrapper ul{
  height: 0;
  opacity: 0;
  padding-right: 1px;
  overflow-y: hidden;
  transition: all 0.2s ease;
}
.wrapper.active ul{
  opacity: 1;
  height: 303px;
}
```

Dictionary/script.js

```
const wrapper = document.querySelector(".wrapper"),
searchInput = wrapper.querySelector("input"),
volume = wrapper.querySelector(".word i"),
infoText = wrapper.querySelector(".info-text"),
synonyms = wrapper.querySelector(".synonyms .list"),
removeIcon = wrapper.querySelector(".search span");
let audio;
function data(result, word){
    if(result.title){
        infoText.innerHTML = `Can't find the meaning of <span>"${word}"</span>. Please,
try to search for another word.`;
```

```
  }else{
      wrapper.classList.add("active");
      let definitions = result[0].meanings[0].definitions[0],
      phontetics = `${result[0].meanings[0].partOfSpeech}  /${result[0].phonetics[0].text}/`
      document.querySelector(".word p").innerText = result[0].word;
      document.querySelector(".word span").innerText = phontetics;
      document.querySelector(".meaning span").innerText = definitions.definition;
      document.querySelector(".example span").innerText = definitions.example;
      audio = new Audio(result[0].phonetics[0].audio);
      if(definitions.synonyms[0] == undefined){
         synonyms.parentElement.style.display = "none";
      }else{
         synonyms.parentElement.style.display = "block";
         synonyms.innerHTML = "";
         for (let i = 0; i < 5; i++) {
            let tag = `<span onclick="search('${definitions.synonyms[i]}')">$
{definitions.synonyms[i]},</span>`;
            tag = i == 4 ? tag = `<span onclick="search('${definitions.synonyms[i]}')">$
{definitions.synonyms[4]}</span>` : tag;
            synonyms.insertAdjacentHTML("beforeend", tag);
         }
      }
   }
}
function search(word){
   fetchApi(word);
   searchInput.value = word;
}
function fetchApi(word){
   wrapper.classList.remove("active");
   infoText.style.color = "#000";
   infoText.innerHTML = `Searching the meaning of <span>"${word}"</span>`;
   let url = `https://api.dictionaryapi.dev/api/v2/entries/en/${word}`;
   fetch(url).then(response => response.json()).then(result => data(result, word)).catch(()
=>{
      infoText.innerHTML = `Can't find the meaning of <span>"${word}"</span>. Please,
try to search for another word.`;
   });
}
searchInput.addEventListener("keyup", e =>{
   let word = e.target.value.replace(/\s+/g, ' ');
   if(e.key == "Enter" && word){
      fetchApi(word);
   }
});
```

```
volume.addEventListener("click", ()=>{
    volume.style.color = "#4D59FB";
    audio.play();
    setTimeout(() =>{
        volume.style.color = "#999";
    }, 800);
});
removeIcon.addEventListener("click", ()=>{
    searchInput.value = "";
    searchInput.focus();
    wrapper.classList.remove("active");
    infoText.style.color = "#9A9A9A";
    infoText.innerHTML = "Type any existing word and press enter to get meaning,
example, synonyms, etc.";
});
```

6. Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of product or service under test.

Software testing is a process used to identify the correctness, completeness and quality of developed computer software. Actually, testing can never establish the correctness of computer software, this can only be done by formal verification. It can only find defects

Why system testing is required?.

1. It is a first level software testing where the software or application is tested as whole.

2. It is done to verify and validate the technical business functional and non-functional requirements of the software. It also include the verification and validation of software application architecture.

Testing Methods

White Box Testing (WBT) :

Entire WBT is done by developers. It is thetesting of each and every line of code in the program.Developers do WBT, sends the s/w to testing team. The testing team does black box testing and checks the s/w against requirements and finds any defects and sends it to the developer.The developers fixes the defect and does WBT and sends it to the testing team.Fixing defect means the defect is removed and the feature is working fine

Grey box testing(GBT) :

It is a mixture of both white box as well as black box testing and it is generally done by the test engineer who has knowledge of both coding and testing

Black box testing(BBT) :

It is a type of testing done by the test engineers where he/she checks if the application(s/w) is working according to the requirement specification.

Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together.Integration testing works to expose defects in the interface and interaction between integrated components.

System Testing

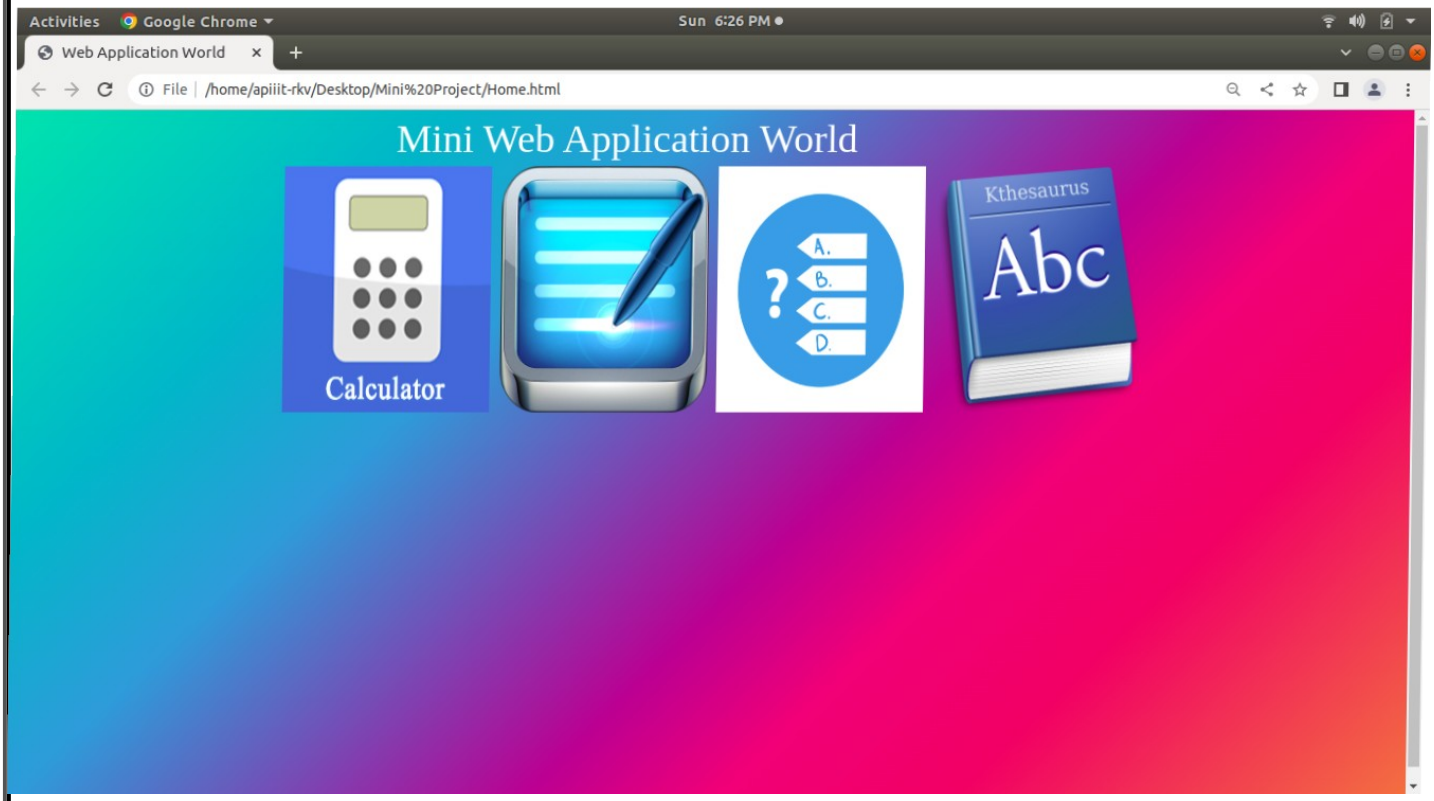
System testing tests a completely integrated system to verify that requirements.

Agile Testing

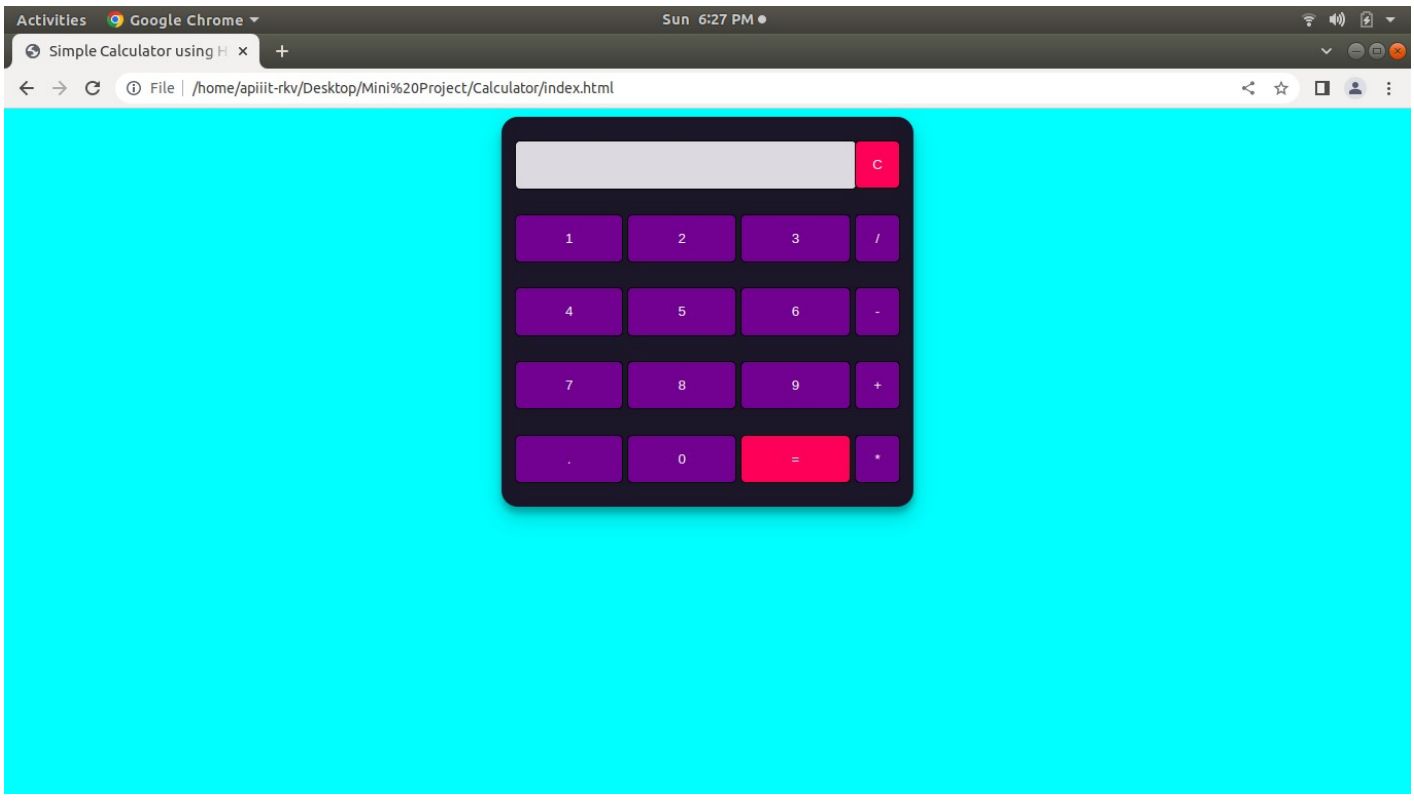
Agile testing is a software testing practice that follows the principles of Agile software development.Agile testing involves all the members of a cross functional agile team, with special expertise contributed by testers, to ensure delivering business value desired by the customer at frequent intervals.

Output

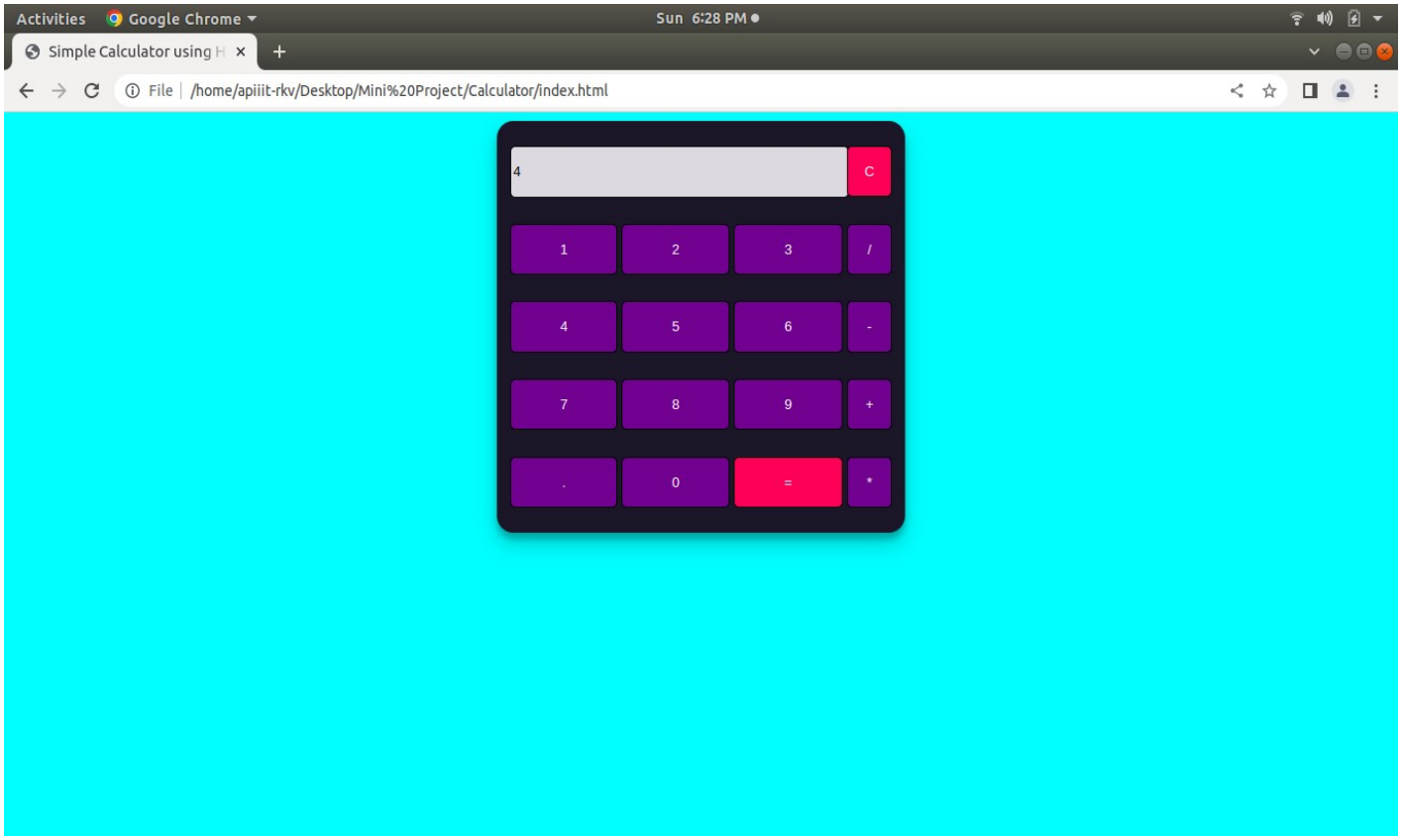
1.To select any application from this click the respective application



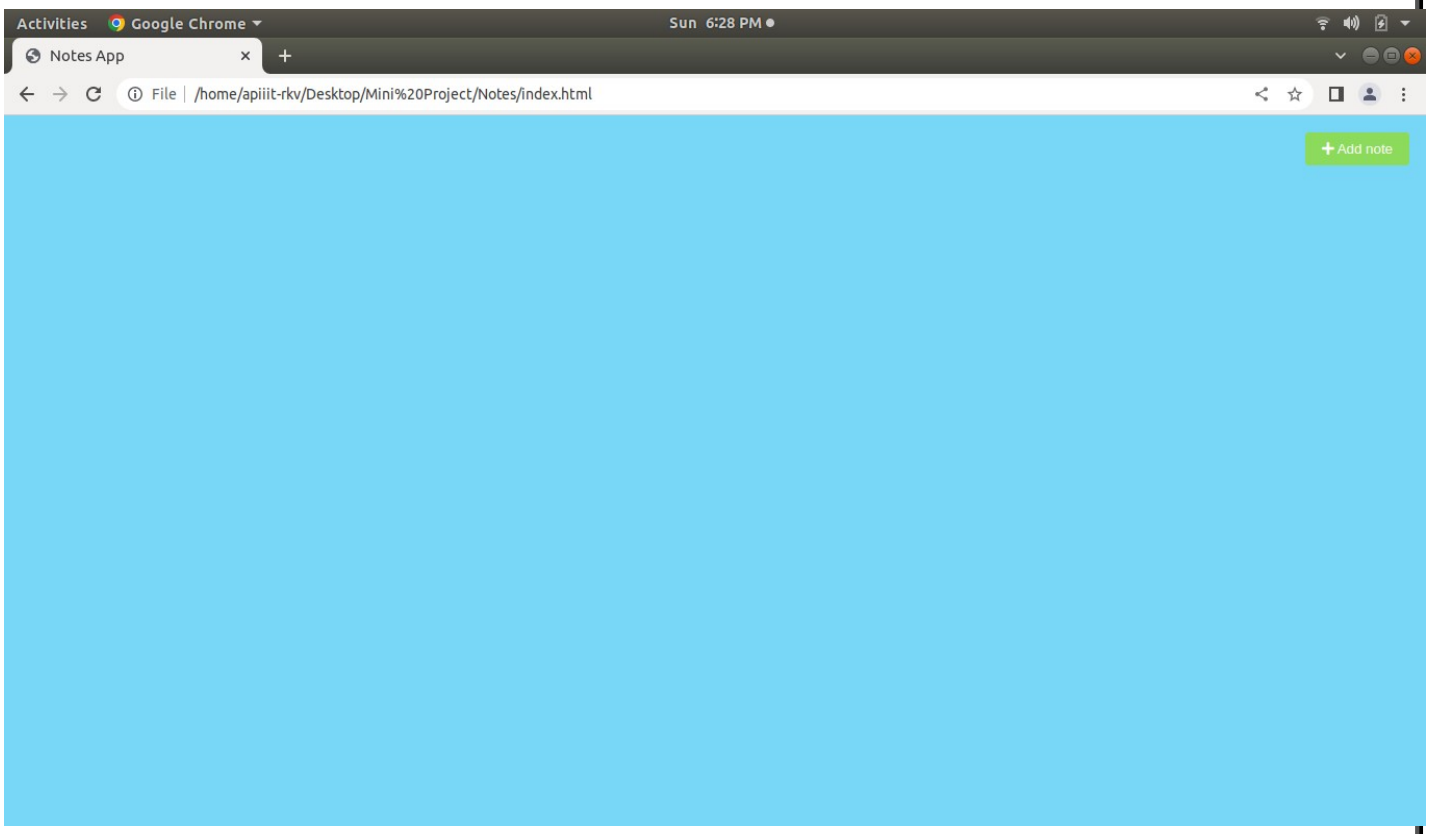
2.After choosing calculator application



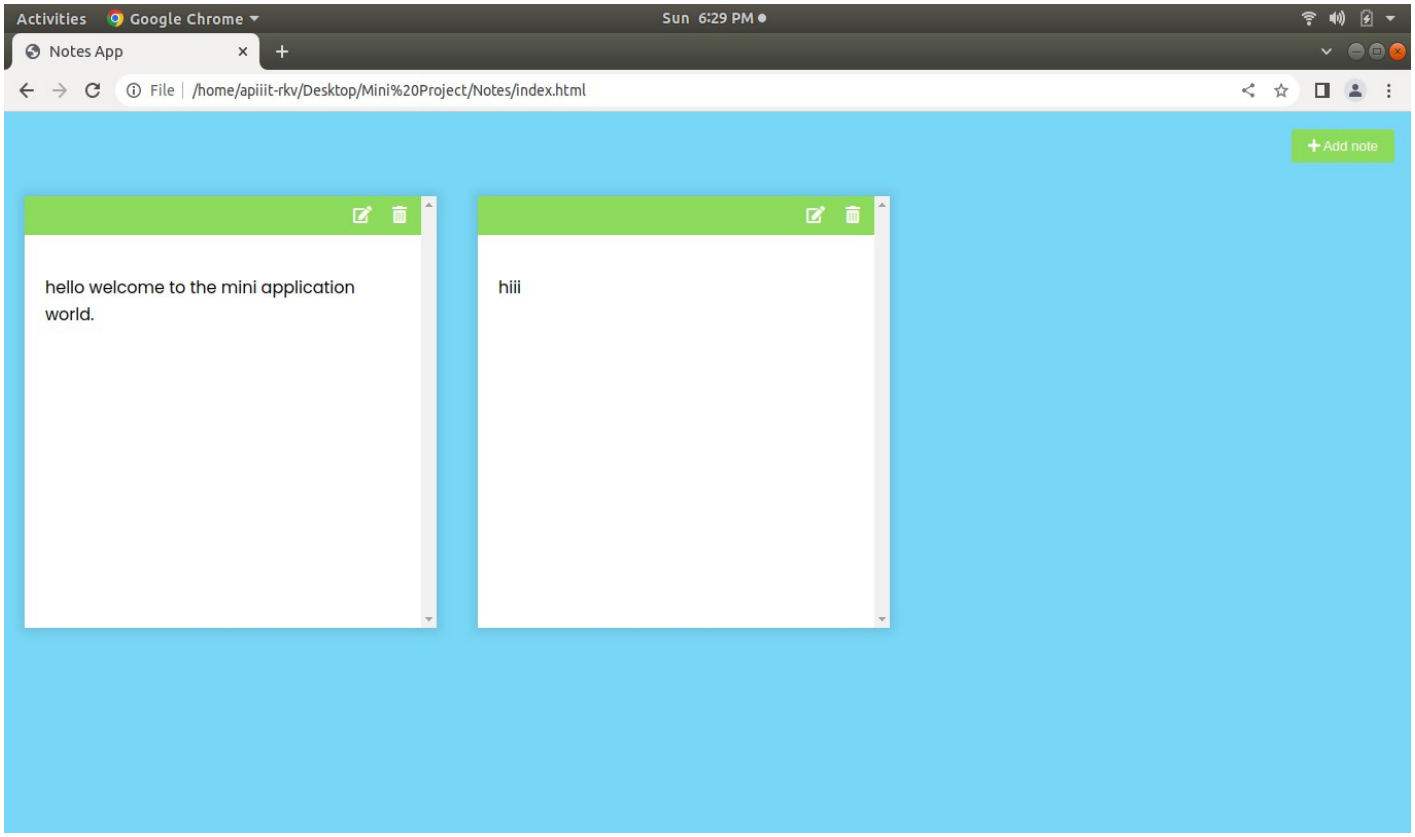
3.After performing addition operation i.e 2+2



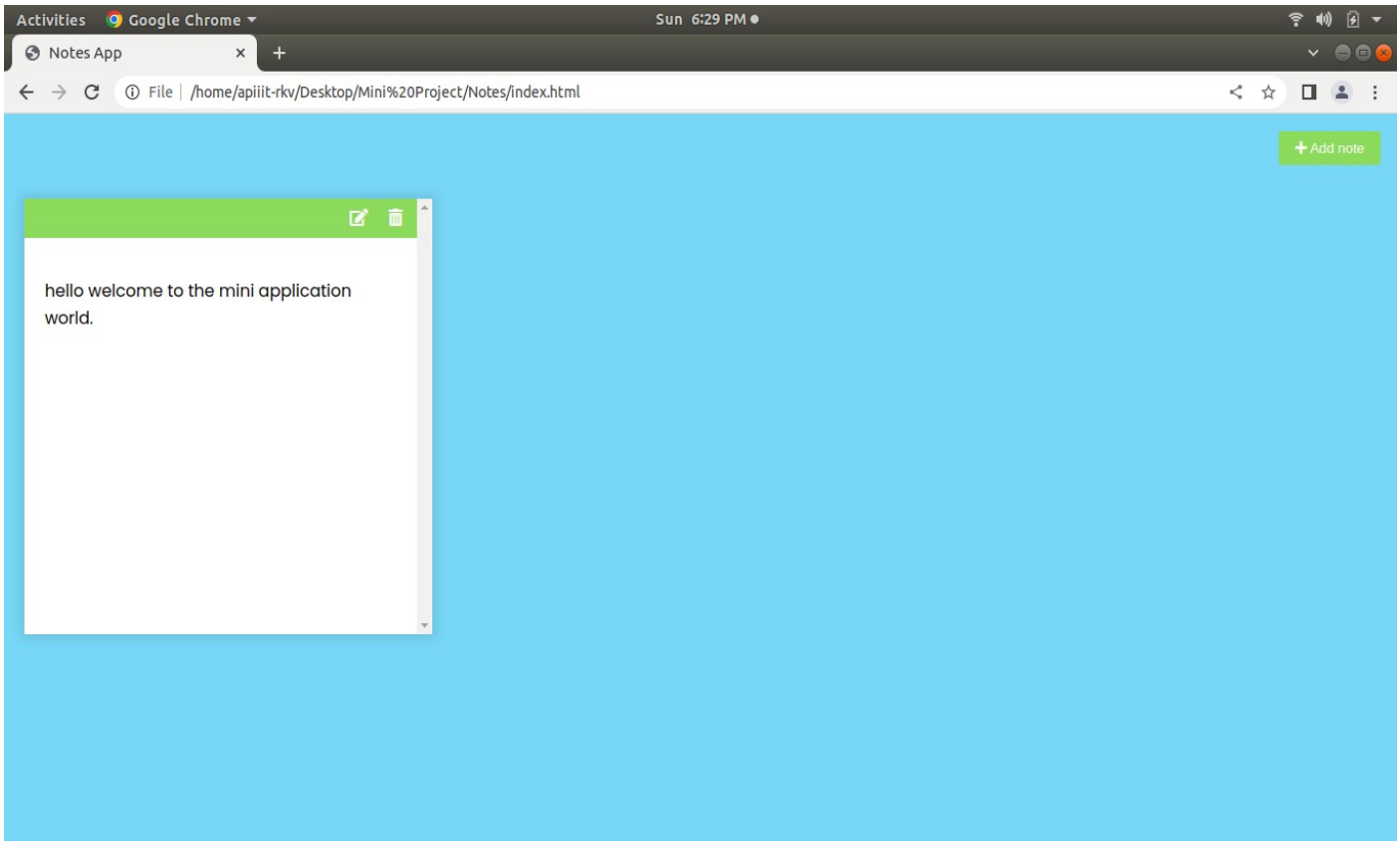
4.After choosing Notes application



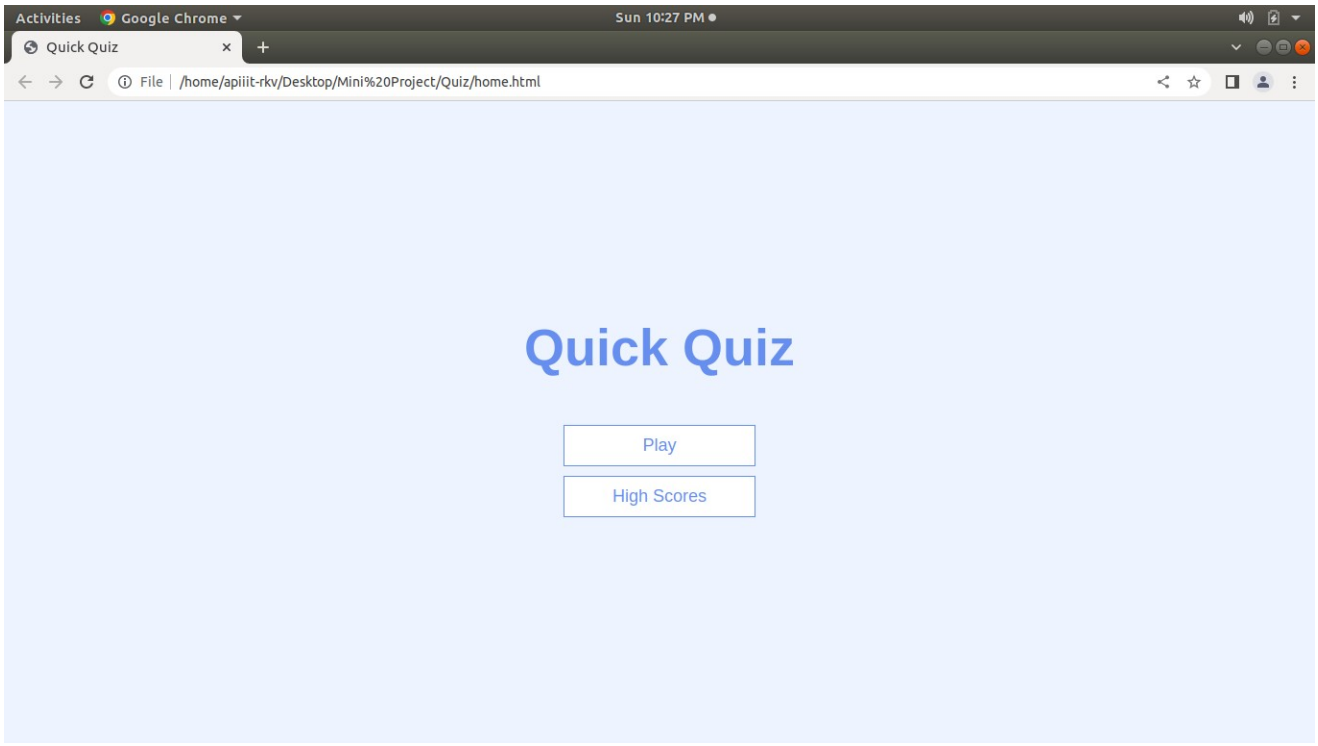
5.After note making in the Notes application



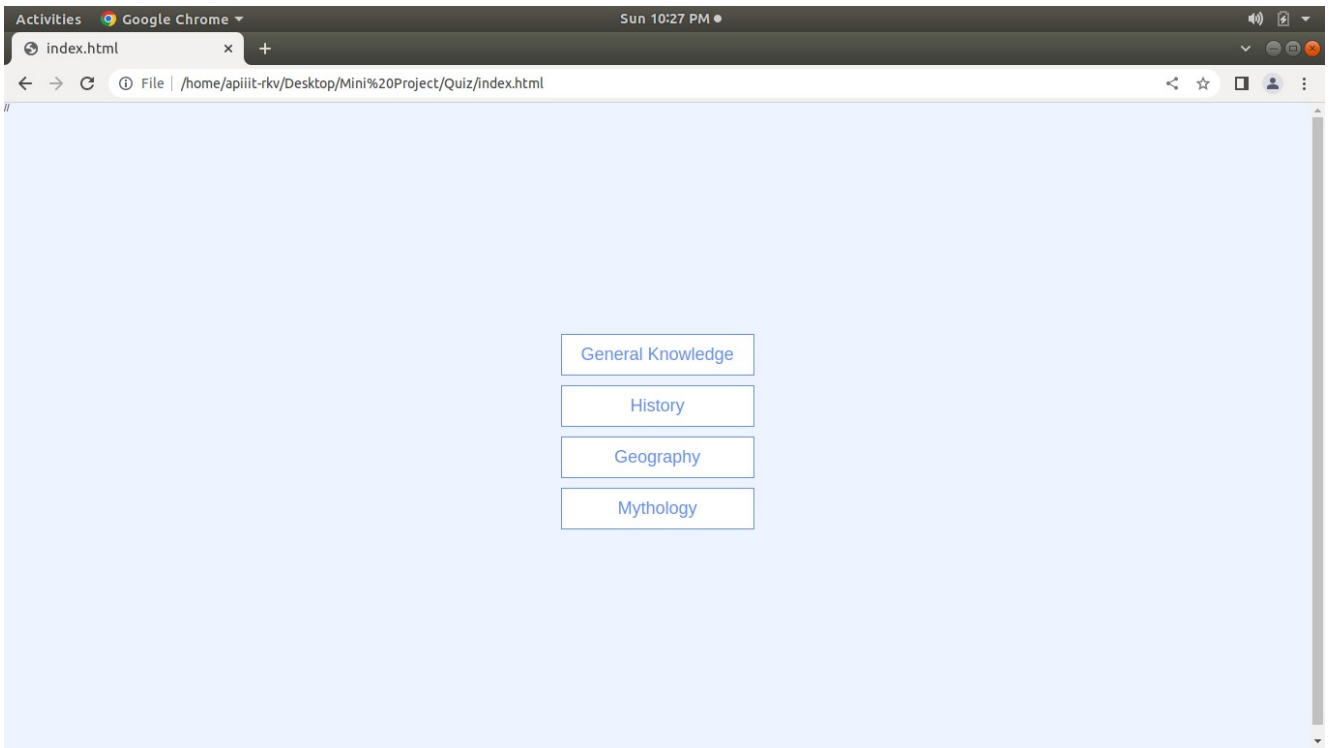
6.After deleting the page from the notes



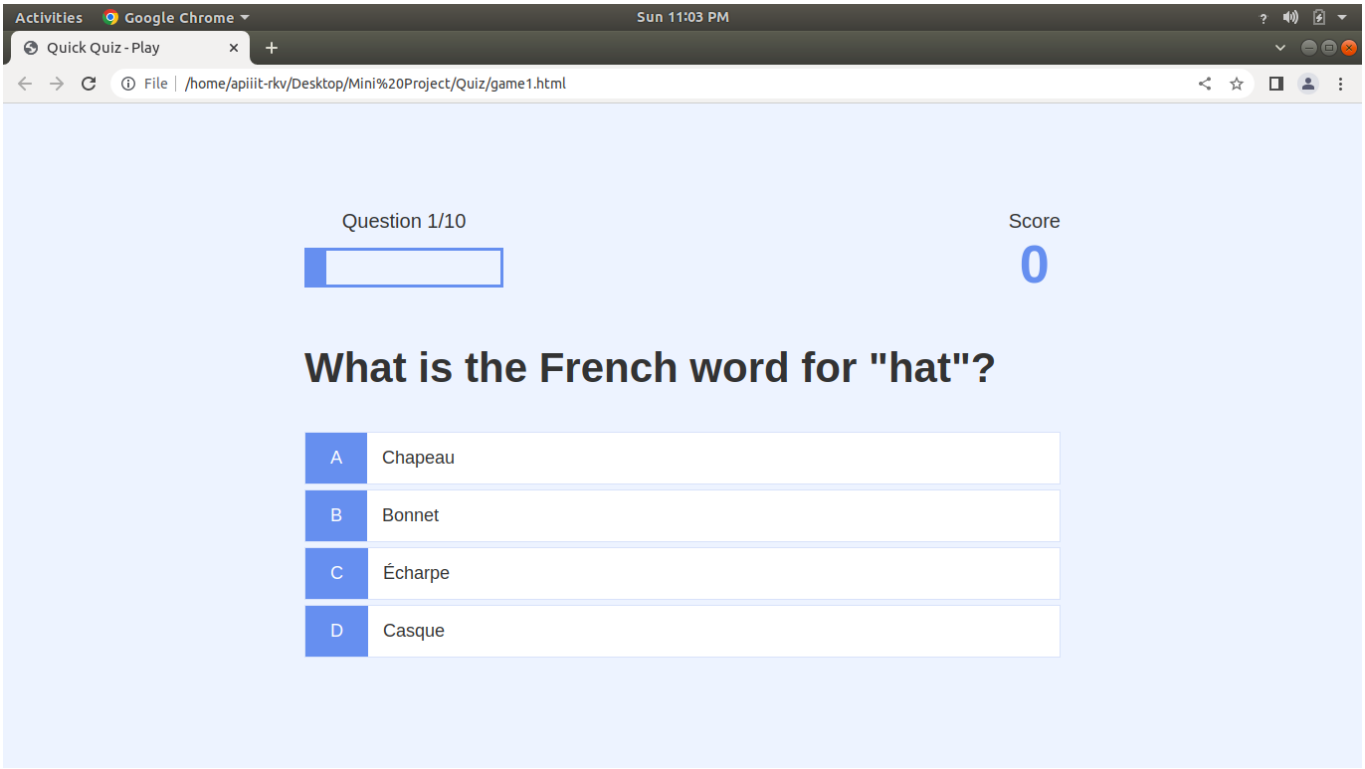
7.After Clicking on Quiz App:



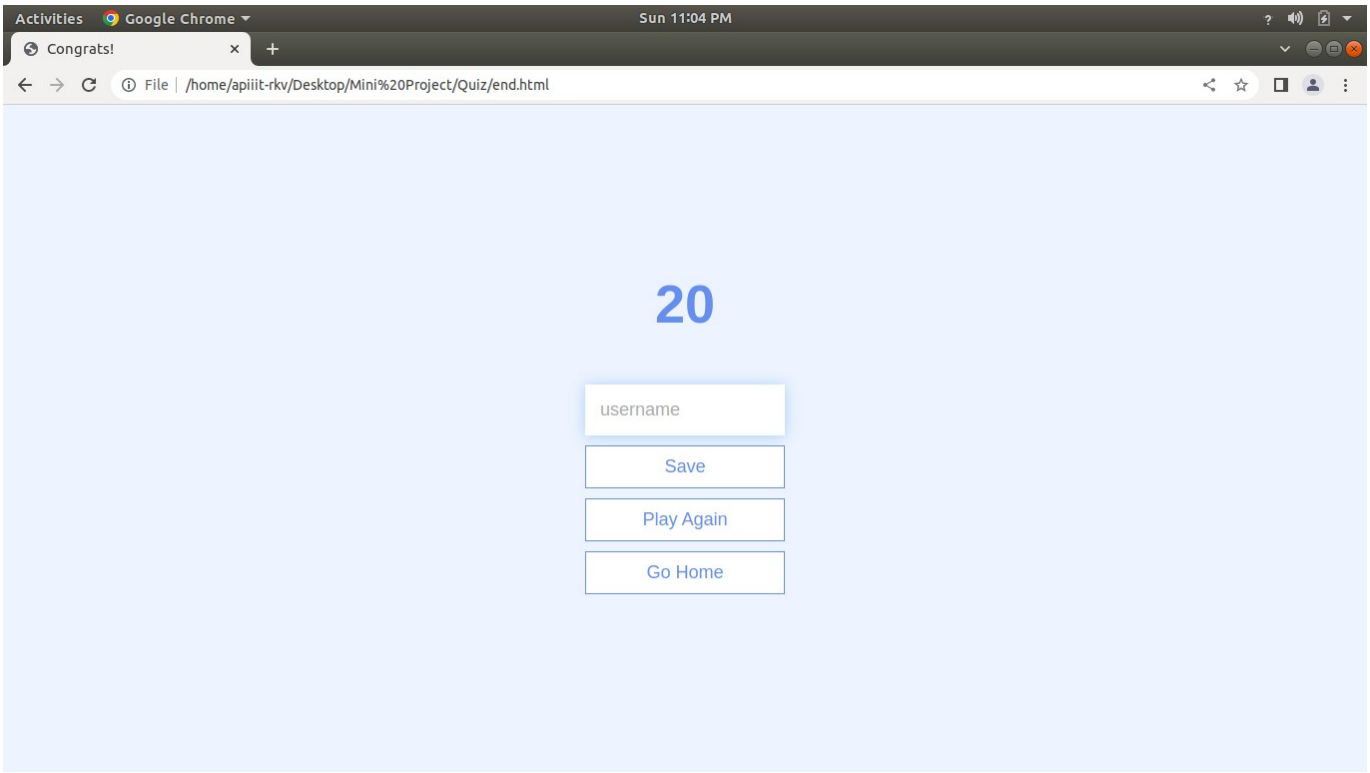
8.After Clicking on Play Button:



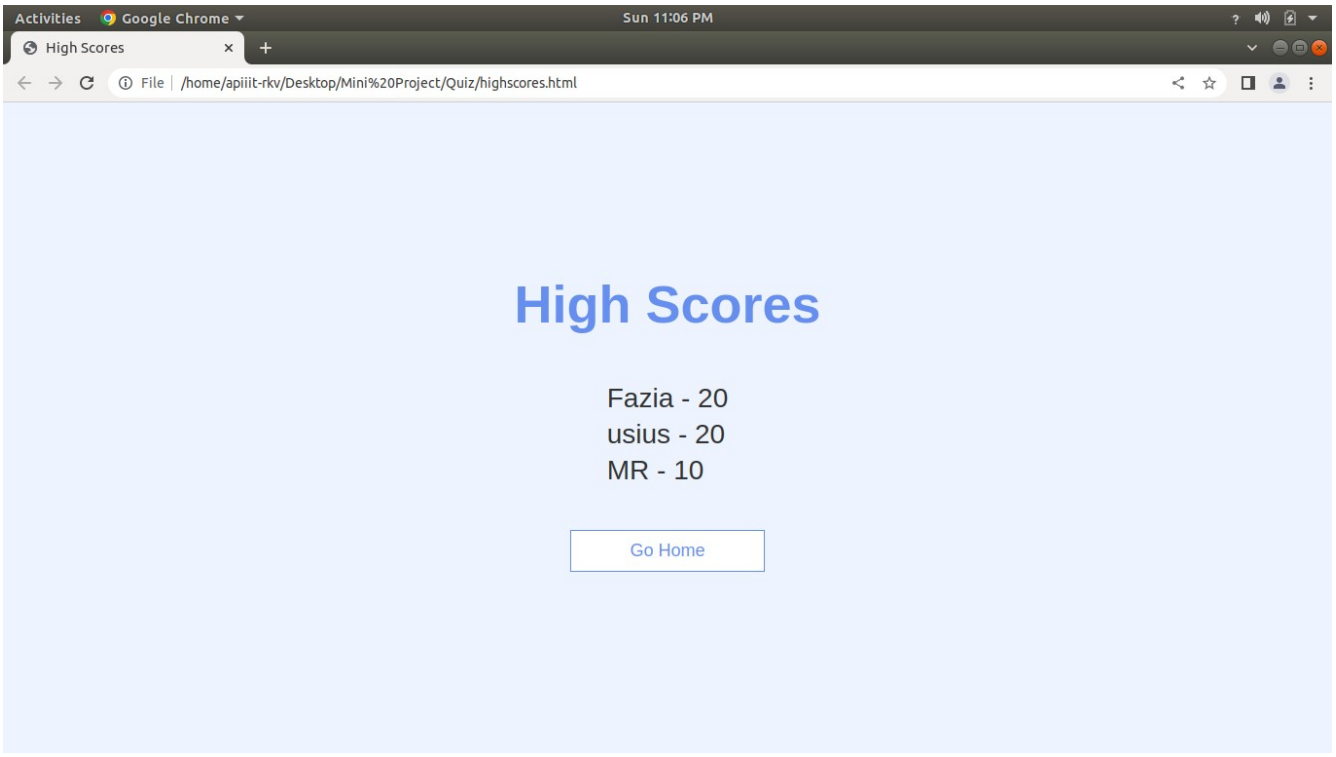
9.After clicking on general knowledge:



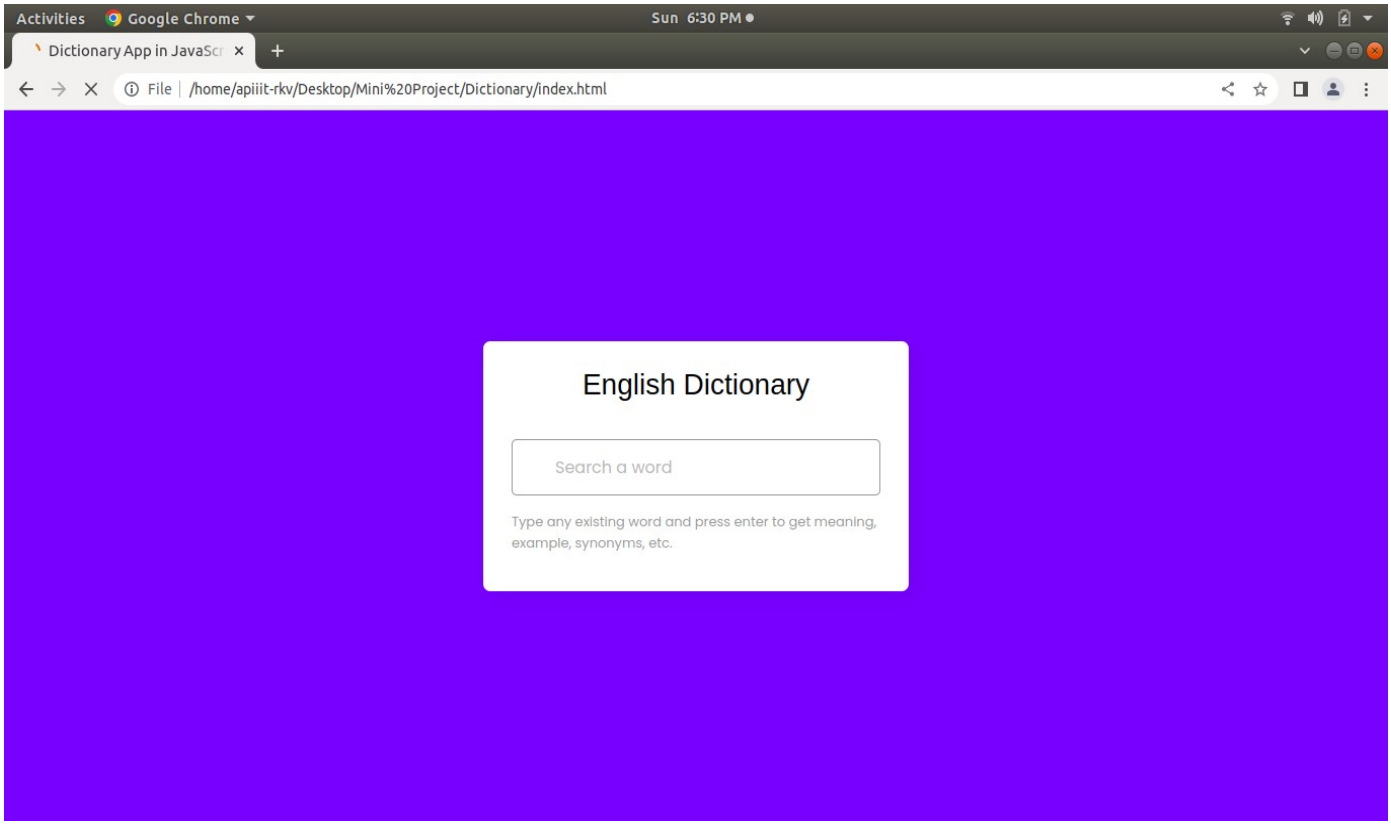
10.After finishing all the questions, present highscore will be displayed:



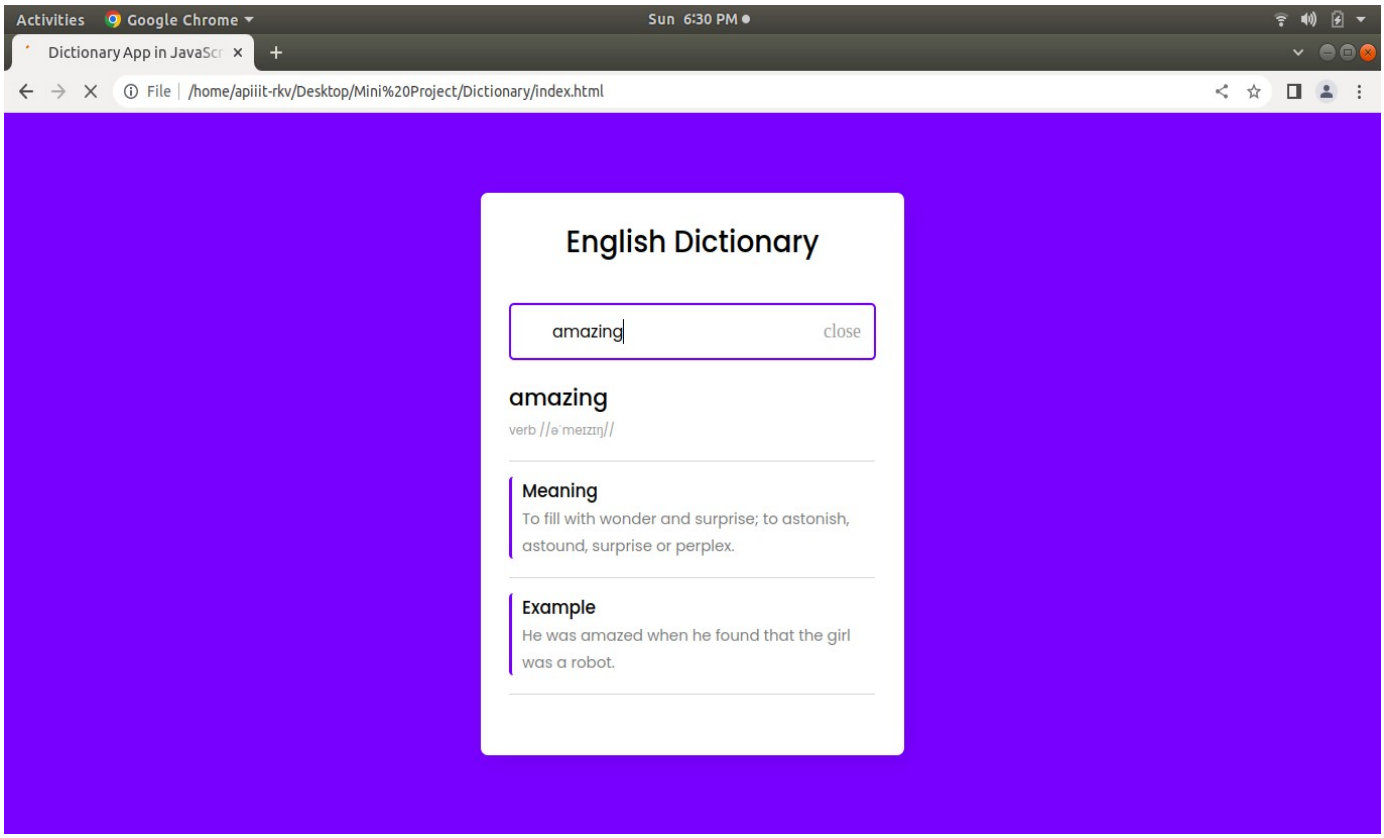
11.After clicking on home and the n Highscores:



11.After choosing Dictionary



12.After searching the meaning of word



Applications:

Mini Application World is a web application that can be used by people irrespective of their age. It is developed in a way such that even a beginner can use it well. We have majorly included 4 sub apps in our main application that are mostly used by people in their everyday life.

The very first one is Calculator Application that is used for computation purposes in almost every situation. The second one is Notes Application that is mainly used to jot down the important information that can be retrieved later when needed. The third one is Quiz Application that mainly focussed aroung General Knowledge, History, Geography and Mythology that helps in enhancing and testing knowledge in those areas. The final one is Dictionary Application that is used to improve our vocabulary.

All these purposeful applications have been integrated into our Mini Application World to serve all our users.

Conclusion

For now we have included these 4 mini applications and based on the usability and need we are looking forward to add few more user friendly applications. As our main motto is to ease student's work, we will be focussing more on satisfying our theme.

References

Foot Note:

1) Javatpoint
2)Trivia Database
3) Github References
4)Youtube

End Note:

1) Team Members