Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Near Jnana Bharathi Campus, Bengaluru-560 056.

(An Autonomous Institution, Aided by Government of Karnataka)

MINI PROJECT REPORT

ON

"WORD HUNTER"

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

BY

SHUBHAM KUMAR SARAS

1DA19CS158

UNDER THE GUIDANCE OF

Mrs. ASHA K N	**Mr. PRAVEENA M V**
Dept. of CSE	**Dept. of CSE**
Dr. AIT	**Dr.AIT**

Department of Computer Science & Engineering

2022-23

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Near Jnana Bharathi Campus, Bengaluru-560 056.
(An Autonomous Institution, Aided by Government of Karnataka)

CERTIFICATE

This is to certify that the project entitled **"WORD HUNTER"** submitted in the partial fulfillment of the requirement of the 7th semester Cloud Computing laboratory curriculum during the year 2022-23 is a result of bonafide work carried out by-

SHUBHAM KUMAR SARAS
1DA19CS158

Signature of the guide

Mrs. ASHA K N	**Mr. PRAVEENA M V**
Dept. of CSE	**Dept. of CSE**
Dr. AIT	**Dr.AIT**

1. Internal Examiner ______________________________

2. External Examiner ______________________________

Dr. Siddaraju,
Head of Department
Department of CSE, Dr.AIT

ACKNOWLEDGEMENT

The sense of contention and elation that accompanies the success of this seminar and the report could be incomplete without mentioning the names of people who have helped me in accomplishing them, peoplewhose constant guidance, support and encouragement resulted in the realization.

I consider myself privileged to express our gratitude and respect towards all those who guided me through the project, **“WORD HUNTER”.**

I take this opportunity to thank, **Dr. M Meenakshi, Principal, Dr. Ambedkar Institute of Technology,** Bengaluru for his support and encouragement.

I am grateful to **Dr. Siddaraju, Head of Department, CSE, Dr. Ambedkar Institute of Technology**, Bengaluru for providing encouragement and support.

I consider ourselves privileged to express our gratitude and respect towards our guide **Mrs. Asha KN** and **Mr. Praveena M V** for constant guidance and support for the completion of the project.

Lastly, I thank all the members of the staff both teaching and non-teaching, friends and last but not the least our parents and family, for helping me directly or indirectly in the completion of the project.

SHUBHAM KUMAR SARAS

1DA19CS158

ABSTRACT

Word Hunt Project is web based application. The main purpose of “WORD HUNTER” is to computerize the Front Office Management of game to develop application which is user friendly, simple, fast and cost-effective.
It is a game where it helps the user to guess the words from the given clue which indeed helps them to improvise their vocabulary skills, english comprehension, usage of different words, knowledge and their ability to learn.
It is a very fun way of learning. The application created is very user friendly, easy and faster. The most efficient application and cost free.

CONTENTS

CHAPTER 6

INTRODUCTION

Word Hunt is a game where the user can play using this application. The task is to guess the correct word based on the given clue where each letter can be guessed from the alphabets. If the alphabet is guessed correctly the alphabet will be placed in the given word. If the alphabet is wrong, then the number of guesses will be reduced.

If all the alphabets are chosen correctly the user wins else the user is lost.

1.1 PROJECT DETAIL

The definition of the project is Word Hunt. This application helps the people to play the game which is free and improvises the vocabulary of the user.

1.2 PROJECT PROFILE

Name of the Project: Word Hunter
Object Description: The purpose of this project is to help the users mostly students to improve their vocabulary skills, knowledge.
Operating System: Windows
Front End: HTML, CSS
Backend: JAVASCRIPT

1.3 PURPOSE:

The main purpose of this project is to improvise the vocabulary skills of the students.It will be even helpful for them to gain the knowledge about different words. One of the fastest and easiest applications to use which is cost free and it can be easily downloaded. It even improves the comprehension and the concentration power of the user.

1.4 PROJECT SCOPE:

Word Hunts are used to enhance students' vocabulary growth. Teachers ask students to look for words and patterns in reading materials based upon selected features.
Word Hunts focus on the structure and meaning of words by turning students'attention to spelling patterns and root words.

1.5 OBJECTIVES:

Opportunities for students to work with words are important to enhancing students'vocabularies, as well as increasing their comprehension.

The Word Hunt strategy is a fun, versatile, and simple technique to improve students' vocabulary. Use this strategy with the whole class, small groups, or individually.

Word Hunts help students learn how words are used in different contexts.

CHAPTER 2

LITERATURE REVIEW

2.1 HTML

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages.

HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

2.3 JAVASCRIPT

JavaScript often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[10] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

2.4 AMAZON WEB SERVICES

AWS Description This module involves utilizing the cloud resources of Amazon Web Services to assist the CNN model to classify input images. The AWS platforms which are in use are S3, Sage Maker, EFS, EC2, Lambda, and API Gateway, which are explained below. Essentially S3 is used to store and prepare the dataset for training. Sage Maker uses the prepped dataset and trains a model that gets stored on EFS. A function to handle user requests is made on AWS Lambda, assisted by API Gateway. Our project uses AWS with the following workflow. First, the compressed dataset is uploaded to S3. We use a notebook instance Sage Maker to unzip the dataset and prepare it for training and upload it back to S3. Then Sage Maker again uses this prepared form of the dataset for training, and stores the trained model in EFS. Lambda configures an API to send the output to the user.

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *Regions* and *Availability Zones*

CHAPTER 3

REQUIREMENT SPECIFICATION

A System Requirements Specification (SRS) (also known as Software Requirement Specification is a document or set of documentation that describes the features and behaviour of a system or software application. It includes a variety of elements (see below) that attempts to define the intended functionality required by the customer to satisfy their different users. The hardware and software components of a computer system that are required to install and use software efficiently are specified in the SRS. The minimum system requirements need to be met for the programs to run at all times on the system.

3.1 HARDWARE REQUIREMENTS

Usage of CPU, RAM and storage space can vary significantly based on user behaviour. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active users are. The hardware requirement specifies the necessary hardware which provides us the platform to implement our programs

- Processor: Core i7 8750H.
- Processor Speed: 1.4 GHZ or above.
- RAM: 4 GB RAM or above.
- HARD DISK: 20 GB hard disk or above.

The hardware and software components of s computer system that are required to install and use software efficiently are specified in SRS. The minimum system requirements need to be met for the program to run all times on the system.

3.2 SOFTWARE REQUIREMENTS

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document.

The software requirements specify the pre-installed software needed to run the code being implemented in this project.

- Operating System: Windows 10, MacOS 10.15
- Backend Software: JavaScript
- Frontend Software: HTML, CSS
- AWS EC2

CHAPTER 4

DESIGN

4.1 HTML

The **Hyper Text Markup Language** or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages.

HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

4.2 CSS

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

4.3 JAVASCRIPT

JavaScript often abbreviated as **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[10] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

CHAPTER 5

IMPLEMENTATION

HTML CODE :
index.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8" />
    <title>Guess a Word Game JavaScript</title>
    <link rel="stylesheet" href="style.css" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  </head>
  <body>
    <div class="wrapper">
      <h1>Guess the Word</h1>
      <div class="content">
        <input type="text" class="typing-input" maxlength="1" />
        <div class="inputs"></div>
        <div class="details">
          <p class="hint">Hint: <span></span></p>
          <p class="guess-left">Remaining guesses: <span></span></p>
          <p class="wrong-letter">Wrong letters: <span></span></p>
        </div>
        <button class="reset-btn">Reset Game</button>
      </div>
    </div>
    <script src="js/words.js"></script>
    <script src="js/script.js"></script>
  </body>
</html>
```

The code is a web page with an html document. The code has a head, which includes meta tags and links to style sheets. There is also a title tag that contains the text “guess a word game javascript”.
The body of the code contains two divs: one for the game board and one for the word list. The code is the head of a web page. The code contains meta tags that define the language, character set, and title of the webpage. The code also contains a link to style.css which defines styles for the webpage.
The code is a web page with two divs. The first is the wrapper, which contains the h1 and has an input field for typing in words. The second is content, which has an input field as well but also includes a button to reset the game.
The script tag at the top of this code loads up some javascript files that are needed by this page: “js/words.js” and “js/script.js”. The code is a game that asks the user to guess a word.
The code includes an input field, which will be used for the user to type in their guess. A div with class “content” contains the words that are being guessed and a div with class “inputs” displays all of the letters that have been typed into the input field so far.

CSS CODE :
style.css

```
/* Import Google font - Poppins */
@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap");
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
body {
  display: flex;
  padding: 0 10px;
  min-height: 100vh;
  align-items: center;
  justify-content: center;
  background: #e2dcc8;
}
.wrapper {
  width: 430px;
  background: #fff;
  border-radius: 10px;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
}
.wrapper h1 {
  font-size: 25px;
  font-weight: 500;
  padding: 20px 25px;
  border-bottom: 1px solid #ccc;
}
.wrapper .content {
  margin: 25px 25px 35px;
}
.content .inputs {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
}
.inputs input {
  height: 57px;
  width: 56px;
  margin: 4px;
  font-size: 24px;
  font-weight: 500;
  color: #e2dcc8;
  text-align: center;
  border-radius: 5px;
  background: none;
  pointer-events: none;
  text-transform: uppercase;
  border: 1px solid #b5b5b5;
}
.typing-input {
  opacity: 0;
  z-index: -999;
  position: absolute;
  pointer-events: none;
}
.inputs input:first-child {
  margin-left: 0px;
}
.content .details {
  margin: 20px 0 25px;
}
.details p {
  font-size: 19px;
  margin-bottom: 10px;
}
.content .reset-btn {
  width: 100%;
  border: none;
  cursor: pointer;
  color: #fff;
  outline: none;
  padding: 15px 0;
  font-size: 17px;
  border-radius: 5px;
  background: #e2dcc8;
  transition: all 0.3s ease;
}
.content .reset-btn:hover {
  background: #e2dcc8;
}

@media screen and (max-width: 460px) {
  .wrapper {
    width: 100%;
  }
  .wrapper h1 {
    font-size: 22px;
    padding: 16px 20px;
  }
  .wrapper .content {
    margin: 25px 20px 35px;
  }
  .inputs input {
    height: 51px;
    width: 50px;
    margin: 3px;
    font-size: 22px;
  }
  .details p {
    font-size: 17px;
  }
  .content .reset-btn {
    padding: 14px 0;
    font-size: 16px;
  }
}
```

The code is a wrapper for an h1 element. The wrapper has a width of 430px and is positioned at the bottom left corner of the screen. The background color is #fff, which creates a white background with rounded corners. The border-radius property sets 10px rounded corners on all four sides of the container.
A box shadow is applied to give it depth and make it look like there's something behind it. The code will create a wrapper with a width of 430px and background color of #fff.

The wrapper will have 10px rounded corners and the box-shadow will be 0 10px 25px rgba(0, 0, 0, 0.1). The content inside the wrapper will have a margin of 25px on each side.

Then the code starts with a div that has the class of "inputs". Inside this div, there are two input fields. The first input field is for typing in text and the second input field is for selecting an option from a drop-down menu. The code starts by setting up some basic styles to make sure everything looks nice and neat.

It sets the width of each input field to 56px, which makes them wide enough so that users can type without having to scroll horizontally across their screens.

It also sets the height of each input field to 57px, which means they're tall enough so that users don't have to scroll vertically down their screens when they're filling out forms or typing in text boxes.

Next it adds some margins around each form element so that it doesn't look too crowded on smaller devices like phones or tablets where space might be limited due to screen size constraints.

Finally, it adds a border radius around each form element because borders are cool and people love borders! The code is used to create a simple input field.

Third step to create a word guessing game in html css & javascript is create a javascript file with the name script.js and paste the given codes into your javascript file. Remember, you've to create a file with a .js extension and it should be inside the js folder.

JAVASCRIPT CODE :
script.js

```
const inputs = document.querySelector(".inputs"),
hintTag = document.querySelector(".hint span"),
guessLeft = document.querySelector(".guess-left span"),
wrongLetter = document.querySelector(".wrong-letter span"),
resetBtn = document.querySelector(".reset-btn"),
typingInput = document.querySelector(".typing-input");

let word, maxGuesses, incorrectLetters = [], correctLetters = [];

function randomWord() {
    let ranItem = wordList[Math.floor(Math.random() * wordList.length)];
    word = ranItem.word;
    maxGuesses = word.length >= 5 ? 8 : 6;
    correctLetters = []; incorrectLetters = [];
    hintTag.innerText = ranItem.hint;
    guessLeft.innerText = maxGuesses;
    wrongLetter.innerText = incorrectLetters;

    let html = "";
    for (let i = 0; i < word.length; i++) {
        html += `<input type="text" disabled>`;
        inputs.innerHTML = html;
    }
}
randomWord();

function initGame(e) {
    let key = e.target.value.toLowerCase();
    if(key.match(/^[A-Za-z]+$/) && !incorrectLetters.includes(` ${key}`) && !correctLetters.includes(key)) {
        if(word.includes(key)) {
            for (let i = 0; i < word.length; i++) {
                if(word[i] == key) {
                    correctLetters += key;
                    inputs.querySelectorAll("input")[i].value = key;
                }
            }
        } else {
            maxGuesses--;
            incorrectLetters.push(` ${key}`);
        }
        guessLeft.innerText = maxGuesses;
        wrongLetter.innerText = incorrectLetters;
    }
    typingInput.value = "";

    setTimeout(() => {
        if(correctLetters.length === word.length) {
            alert(`Congrats! You found the word ${word.toUpperCase()}`);
            return randomWord();
        } else if(maxGuesses < 1) {
            alert("Game over! You don't have remaining guesses");
            for(let i = 0; i < word.length; i++) {
                inputs.querySelectorAll("input")[i].value = word[i];
            }
        }
    }, 100);
}

resetBtn.addEventListener("click", randomWord);
typingInput.addEventListener("input", initGame);
inputs.addEventListener("click", () => typingInput.focus());
document.addEventListener("keydown", () => typingInput.focus());
```

- The code is a function that randomly generates a word from the list of words.
- it then creates an input field with text disabled, and it loops through each letter in the word to create an input field for each letter.
- the code is quite simple and easy to understand.
- the code first creates an array of words, then it randomly selects a word from the list.
- the code then creates a text input field with the disabled attribute so that users cannot submit their guesses until they have entered all six letters of the word.
- The code starts by checking if the input is a letter.
- if it is, then the code checks to see if that letter is in the word list.
- if it’s not, then the code increments maxguesses and adds that letter to incorrectletters .
- the code also sets up an alert box with “congrats!

- you found the word" followed by whatever word was chosen at random.
- if correctletters has more letters than words in our list, then we check for whether or not there are any remaining guesses left.
- if there are no remaining guesses left, then we set all of our inputs' values to be equal to their corresponding words in our list (word[i]).
- the code is meant to be executed when the user presses a key on the keyboard.
- the code first checks if the inputted key is in the correct set of letters.
- if it's not, then it will check for an input with that same letter in its name and replace its value with that letter.
- if there are no more guesses left, then it will show an alert saying "game over!"
- and reset all inputs to their original values.

Last step to create a word guessing game in html css & javascript is create a javascript file with the name words.js and paste the given codes into your javascript file. Remember, you've to create a file with a .js extension and it should be inside the js folder. We'll store random word details as an object in this file.

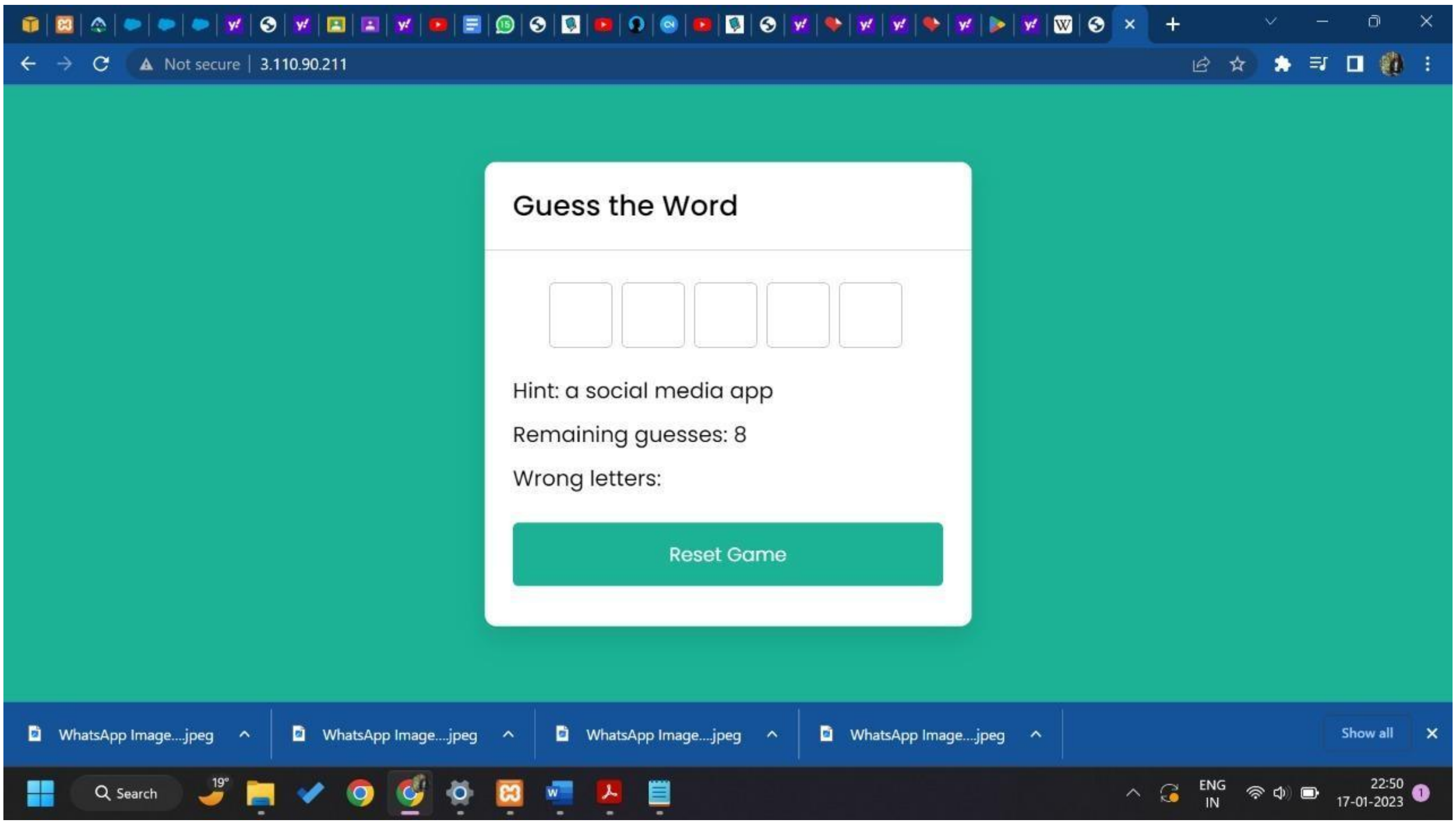
words.js

```
let wordList = [
    {
        word: "python",
        hint: "programming language"
    },
    {
        word: "guitar",
        hint: "a musical instrument"
    },
    {
        word: "aim",
        hint: "a purpose or intention"
    },
    {
        word: "venus",
        hint: "planet of our solar system"
    },
    {
        word: "gold",
        hint: "a yellow precious metal"
    },
    {
        word: "ebay",
        hint: "online shopping site"
    },
    {
        word: "golang",
        hint: "programming language"
    },
    {
        word: "coding",
        hint: "related to programming"
    },
    {
        word: "matrix",
        hint: "science fiction movie"
    },
    {
        word: "bugs",
        hint: "related to programming"
    },
    {
        word: "avatar",
        hint: "epic science fiction film"
    },
    {
        word: "gif",
        hint: "a file format for image"
    },
    {
        word: "mental",
        hint: "related to the mind"
    },
    {
        word: "map",
        hint: "diagram represent of an area"
    },
    {
        word: "island",
        hint: "land surrounded by water"
    },
    {
        word: "hockey",
        hint: "a famous outdoor game"
    },
    {
        word: "chess",
        hint: "related to a indoor game"
    },
    {
        word: "viber",
        hint: "a social media app"
    },
    {
        word: "github",
        hint: "code hosting platform"
    },
    {
        word: "png",
        hint: "a image file format"
    },
    {
        word: "silver",
        hint: "precious greyish-white metal"
    },
    {
        word: "mobile",
        hint: "an electronic device"
    },
    {
        word: "gpu",
        hint: "computer component"
    },
      {
        word: "java",
        hint: "programming language"
      },
      {
        word: "google",
        hint: "famous search engine"
      },
      {
        word: "venice",
        hint: "famous city of waters"
      },
      {
        word: "excel",
        hint: "microsoft product for windows"
      },
      {
        word: "mysql",
        hint: "a relational database system"
      },
      {
        word: "nepal",
        hint: "developing country name"
      },
      {
        word: "flute",
        hint: "a musical instrument"
      },
      {
        word: "crypto",
        hint: "related to cryptocurrency"
      },
      {
        word: "tesla",
        hint: "unit of magnetic flux density"
      },
      {
        word: "mars",
        hint: "planet of our solar system"
      },
      {
        word: "proxy",
        hint: "related to server application"
      },
      {
        word: "email",
        hint: "related to exchanging message"
      },
      {
        word: "html",
        hint: "markup language for the web"
      },
      {
        word: "air",
        hint: "related to a gas"
    },
    {
        word: "idea",
        hint: "a thought or suggestion"
    },
    {
        word: "server",
        hint: "related to computer or system"
    },
    {
        word: "svg",
        hint: "a vector image format"
    },
    {
        word: "jpeg",
        hint: "a image file format"
    },
    {
        word: "search",
        hint: "act to find something"
    },
    {
        word: "key",
        hint: "small piece of metal"
    },
    {
        word: "egypt",
        hint: "a country name"
    },
    {
        word: "joker",
        hint: "psychological thriller film"
    },
    {
        word: "dubai",
        hint: "developed country name"
    },
    {
        word: "photo",
        hint: "representation of person or scene"
    },
    {
        word: "nile",
        hint: "largest river in the world"
    },
    {
        word: "rain",
        hint: "related to a water"
    },
]
```

A WORD HUNTER in HTML, CSS & JAVASCRIPT is now complete. We implemented this with just plain old javascript and used the concept of object-oriented-programming.

CHAPTER 6

RESULTS



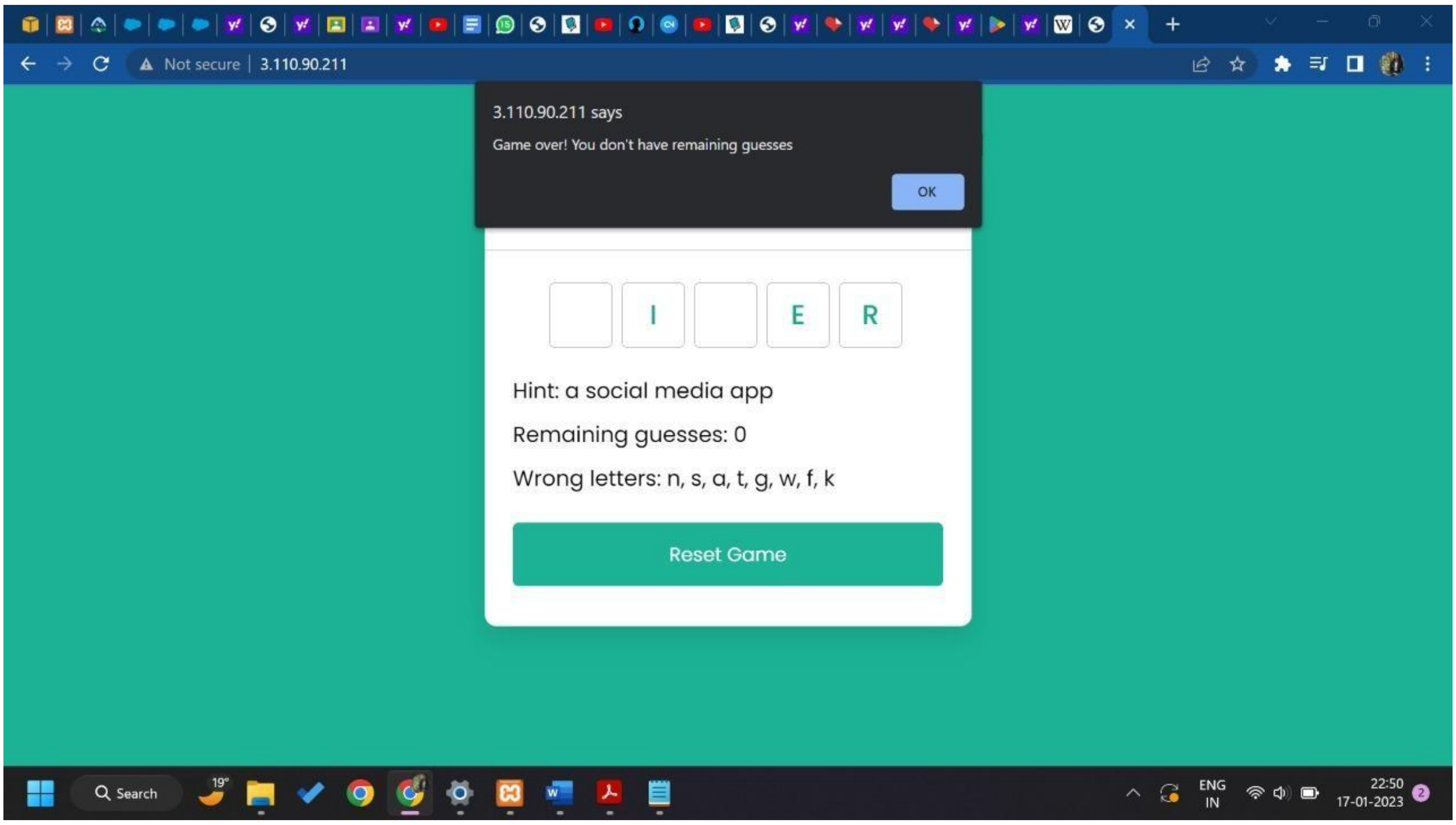
Guess the Word

Hint: a social media app

Remaining guesses: 8

Wrong letters:

Reset Game



3.110.90.211 says

Game over! You don't have remaining guesses

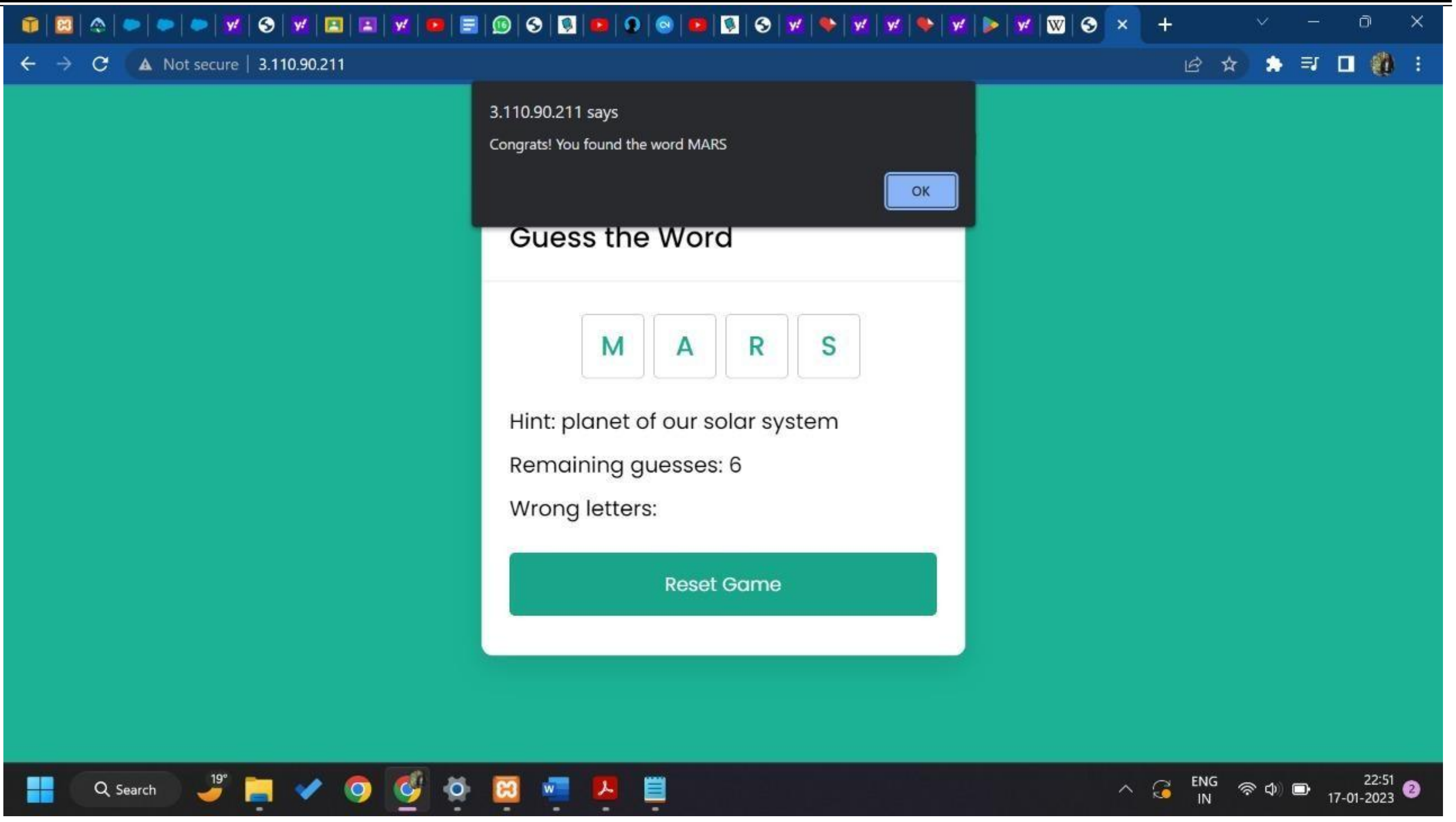
I E R

Hint: a social media app

Remaining guesses: 0

Wrong letters: n, s, a, t, g, w, f, k

Reset Game



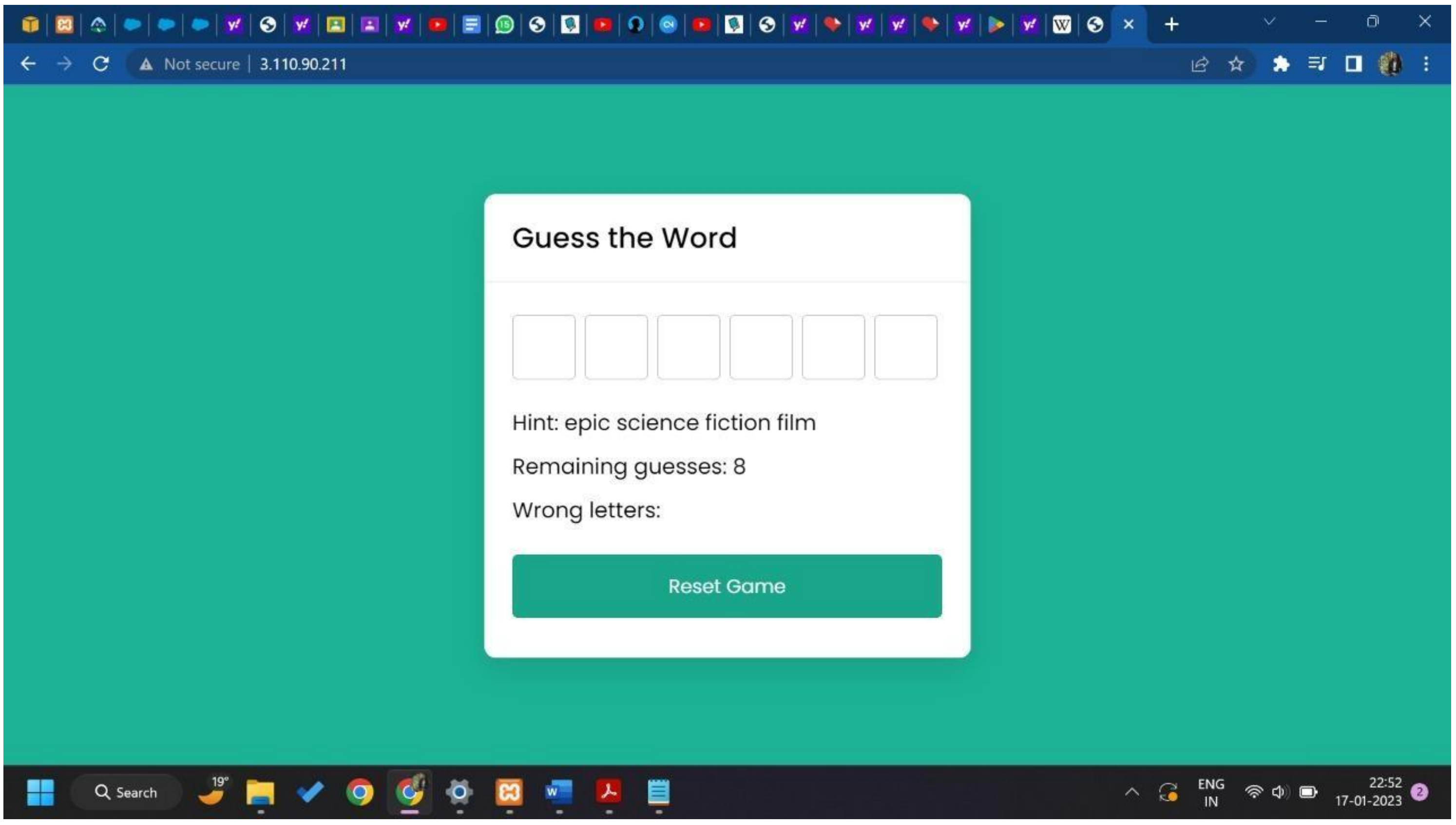
3.110.90.211 says

Congrats! You found the word MARS

Hint: planet of our solar system

Remaining guesses: 6

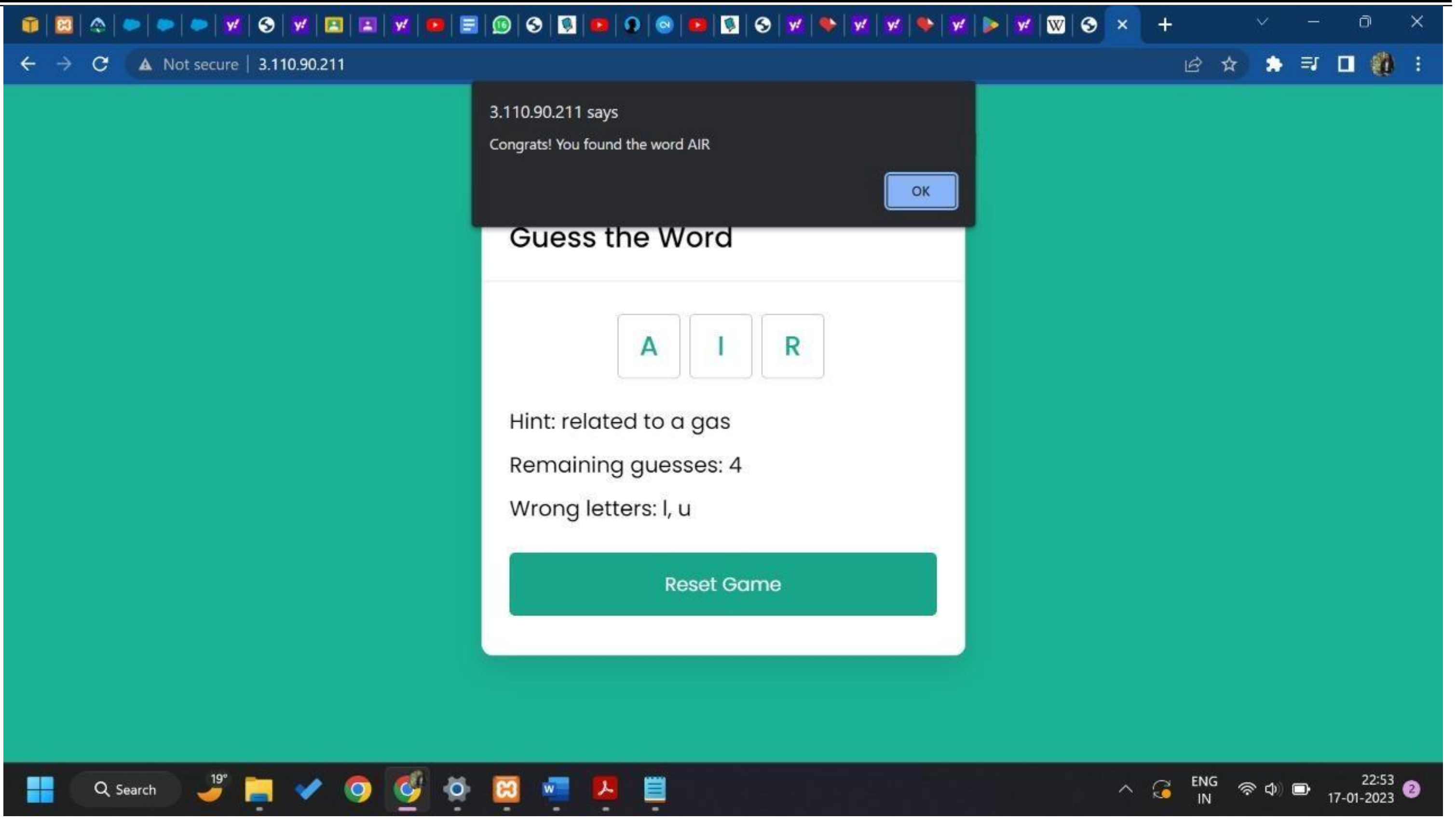
Wrong letters:



Hint: epic science fiction film

Remaining guesses: 8

Wrong letters:



3.110.90.211 says

Congrats! You found the word AIR

Hint: related to a gas

Remaining guesses: 4

Wrong letters: l, u

CHAPTER 7

DEPLOYMENT

Software deployment includes all of the steps, processes, and activities that are required to make a software system or update available to its intended users. Today, most IT organizations and software developers deploy software updates, patches and new applications with a combination of manual and automated processes. Some of the most common activities of software deployment include software release, installation, testing, deployment, and performance monitoring.

Software development teams have innovated heavily over the past two decades, creating new paradigms and working methods for software delivery that are designed to meet the changing demands of consumers in an increasingly connected world. In particular, software developers have created workflows that enable faster and more frequent deployment of software updates to the production environment where they can be accessed by users.

7.1 CLOUD DEPLOYEMENT

Cloud deployment is the process of deploying an application through one or more hosting models—software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing and operating workloads on cloud.

Advantages of Cloud Deployment:

- Faster and simplified deployments. Automate builds that deploy code, databases and application releases, including resource provisioning.
- Cost savings. Control costs using consumption-based pricing and eliminate capex-heavy on-premises environments.
- Platform for growth. Leverage the global infrastructure provided by cloud service providers (CSPs) to seamlessly expand the business into other geographies.
- New digital business models. Exploit the continuous release of features and services by CSPs, incubate new technologies and innovate digital business models.
- Business resiliency. Architect for the availability and fault-tolerance CSPs offer and ensure disaster recovery and business continuity of applications to make the business resilient.
- Agility and scalability. Use autoscaling and scalability to meet peak demands of the business without provisioning for excess capacity.
- Geographic reach. Access applications from any location, on any device, leveraging the connectivity backbone of CSPs.
- Operational efficiency. Use the inherent automation enabled by cloud to increase operational efficiency and reduce human effort.
- A competitive edge. Leverage infrastructure as code and development, security and operations to reduce the time to market for new features and stay ahead of the competition.
- Empowered users. Increase productivity by empowering users with self-service options on cloud, such as portals, DevOps pipelines, and executive and operational dashboards.

For our application we will be using Google App Engine for deployment of the application which provides a platform as a service to host our applications

7.2 Deploying on Amazon Web Service's EC2 Instance :

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.
Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

EC2 Advantages :

1. You don't require any hardware units
2. Easily scalable (up or down)
3. You only pay for what you use
4. You have complete control
5. Highly secure
6. You can access your assets from anywhere in the world

CONCLUSION

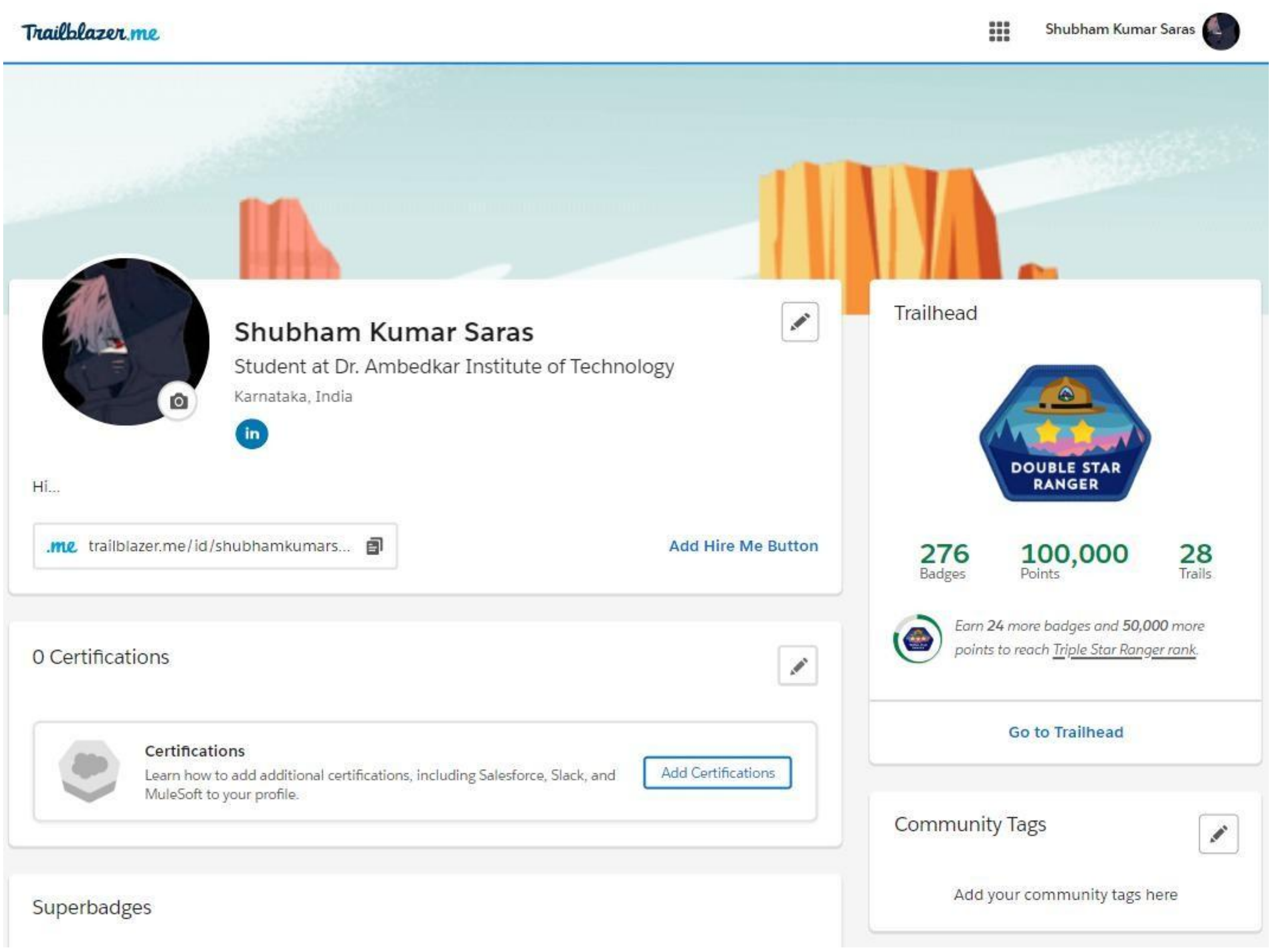
The purpose of the project entitled **“WORD HUNTER”** is to computerize the Front Office Management of game to develop application which is user friendly, simple, fast and cost- effective. It is a game where it helps the user to guess the words from the given clue which indeed helps them to improvise their vocabulary skills, English comprehension, usage of different words, knowledge and their ability to learn. It is a very fun way of learning. The application created is very user friendly, easy and faster. The most efficient application and cost free.

BIBLIOGRAPHY

URL:

1. www.google.com
2. http://www.stackoverflow.com3.www.w3schools.com
3. www.projecttopics.info
4. www.wikipedia.com
5. www.quora.com
6. http://www.geeksforgeeks.org
7. https://docs.aws.amazon.com/efs/latest/ug/gs-step-one-create-ec2-resources.html
8. https://www.adlit.org/in-the-classroom/strategies/word-hunts
9. https://www.codewithrandom.com/2022/08/11/word-guessing-game-in-javascript/
10. https://www.codingnepalweb.com/word-guessing-game-html-css-javascript/

INDUSTRY CERTIFICATION IN SALESFORCE



Trailblazer.me

Shubham Kumar Saras

Shubham Kumar Saras

Student at Dr. Ambedkar Institute of Technology

Karnataka, India

Hi...

.me trailblazer.me/id/shubhamkumars...

Add Hire Me Button

0 Certifications

Certifications

Learn how to add additional certifications, including Salesforce, Slack, and MuleSoft to your profile.

Add Certifications

Superbadges

Trailhead

DOUBLE STAR RANGER

276 Badges 100,000 Points 28 Trails

Earn 24 more badges and 50,000 more points to reach Triple Star Ranger rank.

Go to Trailhead

Community Tags

Add your community tags here