NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

Golden Jubilee
YEARS OF EXCELLENCE IN EDUCATION

A MINI PROJECT REPORT

on

"Hire Hub"

Submitted by

DIVYA.M, EKKURTHI SAI BHAVANA

USN: 1NH19IS042,1NH19IS043

Under the guidance of,

Mr. Saravanan

Sr. Assistant Professor, ISE, NHCE

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

FOR

COURSE NAME :MINI PROJECT

COURSE CODE :20ISE59A

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

50 Golden JUBILEE
YEARS OF EXCELLENCE IN EDUCATION

CERTIFICATE

Certified that the project work entitled Hire-Hub carried out by Ms. Divya M and Ekkurthi Sai Bhavana bearing USN 1NH19IS042 1NH19IS043 bonafide students of V semester in partial fulfillment for the award of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya Technological University, Belagavi during the year 2021-22. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.

Name & Signature of Guide	**Name & Signature of HOD**	**Name & Signature of Principal**
Mr. Saravanan	Dr.Anandhi R J	Dr. Manjunatha

Examiners :

Name **Signature**

1. ………………………………………………… ……………………………………….

2. ………………………………………………… ……………………………………….

ACKNOWLEDGEMENT

Any project is a task of great enormity and it cannot be accomplished by an individual without support and guidance. I am grateful to a number of individuals whose professional guidance and encouragement has made this project completion a reality.

I have a great pleasure in expressing my deep sense of gratitude to the beloved Chairman **Dr. Mohan Manghnani** for having provided me with a great infrastructure and well-furnished labs.

I take this opportunity to express my profound gratitude to the Principal **Dr. Manjunatha** for his constant support and management.

I am grateful to **Dr. R J Anandhi** , Professor and Head of Department of ISE, New Horizon College of Engineering, Bengaluru for his strong enforcement on perfection and quality during the course of my project work.

I would like to express my thanks to the guide **Mr. Saravanan**, Senior Assistant Professor, Department of ISE, New Horizon College of Engineering, Bengaluru who has always guided me in detailed technical aspects throughout my project.

I would like to mention special thanks to all the Teaching and Non-Teaching staff members of Information Science and Engineering Department, New Horizon College of Engineering, Bengaluru for their invaluable support and guidance.

DIVYA.M, EKKURTHI SAI BHAVANA

1NH19IS042, 1NH19IS043

TABLE OF CONTENTS

LIST OF FIGURES

Abstract

Hire Hub as the name suggests, it is a website related to placements. This is an idea to guide the fresher's and to prevent unemployment. It would have certain sign in and would check the data to connect any user to the respective page.

There would be two different profiles to create the view for the respective users. A student profile would guide the fresher's about the process of hiring by providing aptitude test, verbal test, coding test and also a flash card material to help the students brush up with the important formulae needed for aptitude. A placement profile will be handled by the placements head to keep an update over the students being placed and not being placed. If the students are not placed then the placements department could provide some essential guidelines to help the students crack the interview.

The scripting languages- HTML, CSS, Javascript, SQL are used to develop this Hire Hub website. It would be an informative website.

CHAPTER 1

1 INTRODUCTION

Hub is like a central informative area where all the essential information remains centralized.

We came up with the name Hire Hub to show the importance of the website. Here we provide all the preparation material in one place and also give access to placement head to keep track of the progress in the placement of students.

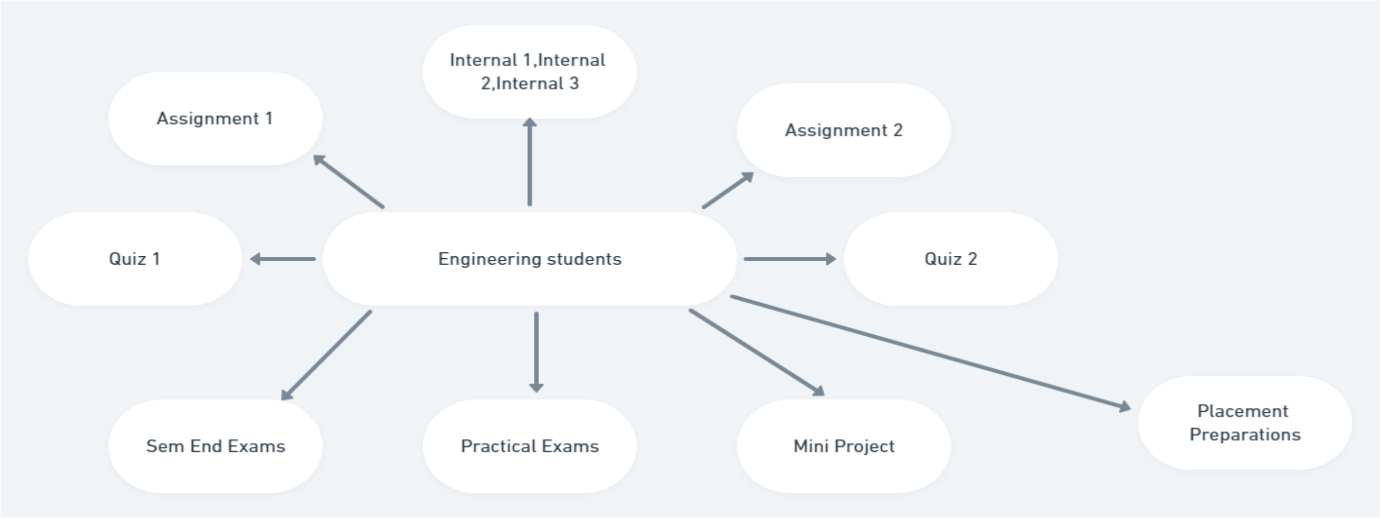


Fig-1 illustration of responsibility

Students are usually pressurized with lots of deadlines and this website would serve well for the last minute preparations. Usually students tend to give up at last moment not knowing what to revise and what not to revise. This website handles this dilemma just like the icing on the cake. It provides an essential guide by providing important frequently asked questions and also the clear solutions to understand every type of problem.

1.1 Motivation Of Project

As mentioned earlier an engineering student has got lots to deal with for each semester. The full 6 months is filled with one or the other task. Making time for placement preparation on this busy schedule is hectic for any student. A student might make time for placement preparation but it might not work continuously as the student has to deal with exams every one month. Preparation with coding and aptitude got to be continuous to be confident enough. This website provides an hand of help for last minute tensionless preparation.

The aim is to provide all the essential materials on a single site. Usually students face this confusion as to which site to refer to get the best practice on different type of questions. We provide a clear solution to that dilemma on our website. All types of questions are covered with clear solutions .This site has an additional benefit of practice as the questions are taken from the previous year’s placement companies. This would help the students to hope and practice higher to get placed in one of the best companies with a high package.

This website aims to boost the confidence of every student before placements. It also encourages students to become perfect with the solutions by giving them an opportunity to take retest over the same set of questions. It provides a clear updated information about the status of placements to the placement coordinator. This helps the coordinator to keep track of students who are not placed and can lend an additional help to get placed in any of the open companies, by training them to the best of their ability.

1.2 Problem Statement

We need to create two different views for 2 different type of users. One for the students and the other for the placements head. To give a clear clarity on the process of placements to the fresher's we need to provide an efficient guide material with all possible type of questions, We should provide clear solutions for them to learn from their mistakes. We should also provide a formulae material for the students to brush up on their aptitude topics. All these must be the features of a student view. Another view must be created for the placement department. It should be accessable only to the placements-head. It should provide the details of all students and their placement status. Here the placement-head should be able to view and update the status of each student. The sign-up and sign-in validations should be made to both the views. The website should be developed to work on real data. Hence the resultant website should be completely informative to both the users.

CHAPTER 2

2 Literature Survey

The main purpose of such a website is to help the students prepare well for their interviews. We provide a proper preparation kit which would be of great help to all the last minute preparative students. The idea was to provide every possible help on a single website. We know that we have separate sites for everything like CodeChef for coding, GeeksForGeeks site for aptitude and verbal etc. The Main aim of this hire hub is to centralize every possible questions with solutions on a single website. This would be very useful as the student need not open different sites to prepare different type of questions. This would help the student save time on preparation and would help the student gain confidence to perform better.

This was developed to help a student prepare tensionless. In simple terms it is a student friendly site. It also helps the placement-head to keep track over the status of placements. Placement department can provide additional help to students who are a little weak at aptitude and coding. This can ensure 100 percent placements and also can help the college gain a reputed name. It is mainly developed to prevent unemployment. There is a saying that not every engineering students will be hired due to hectic competition between different fields of engineering and also between colleges. To help the students handle these situations with calm and composed mindset we decided to come up with this wonderful informative website called Hire Hub.

2.1 Existing System

Every possible preparation kit is available on internet. The only requirement to access all possible information is to iterate over different sites. When thousand such sites are available students tend to get confused as to which site to follow. For example we have CodeChef, HackerRank, TopCoder etc working for the same purpose. These sites help the students to build their coding skills providing them various levels of questions and they also conduct various hackathon to test the time constraints involved in problem solving. The same goes with aptitude and verbal websites. We get thousand such suggestions and each site has its own benefits. Choosing the site wisely would require visiting all the site. We breakdown the efforts of figuring out which site is the best for preparation by providing the best type of questions with clear solutions on a single website.

2.2 Proposed System

We can overcome the drawbacks of existing system by providing additional features. We can give all type of questions to user with time constraint, we can conduct competitive rounds which would help the students understand the level of competition that exists to get hired. We can make it a little more complex by enabling students to work on the website from all over the world. Here the ranking of students will help them boost their confidence and will also help the students understand their strength and weakness.

2.3 Objectives Of Proposed System

Hire hub provides an additional help of time management. Here the students will get to know how many questions they are able to solve within the stipulated time. The students are also given an opportunity to take up retest so that they can improve their thinking capability over the same set of questions. A formulae card is provided as an additional benefit, this would help the students to brush up on the important formulae of each topic and use of these formulae will help them solve every type of aptitude questions. It helps the placement team to have a clear track and progress over the placements status. Here the placement head can access the information and also edit and update the status of each student.

CHAPTER 3

3 System Requirement Specification

We require HTML and CSS for the frontend development of the website. We have two views for the respective users. The page which is seen by the user to see their respective page is completely done using frontend tools. The redirection and validations of sign in is done by JS the backend development tool. We have given nice cards which would flip on hover on student's access page. Each card has its unique operation. First card displays the formulae that is essential to solve aptitude questions. Second card displays the coding questions, when we enter it we get to see three different level of coding with small description about the levels and motivating quote. The user can choose any level of his/her preference and start coding it. The code is provided under each question incase the user feels he/she is not confident enough to solve the problem by oneself they can use it to learn the logic.

The third card holds the verbal questions for practice, on entering this card the user will be able to see the top 5 topics of verbal questions with a small description about each topic on their respective cards. The user can choose the topic of their interest the test rules are displayed describing the rules of the test. For verbal we hardly require 20 seconds so the time limit encoded per question is 20 seconds. The right answer for each question will be displayed soon after the 20 seconds time limit or as soon as the user chooses the wrong answer. If the answer chosen by the user is right then a green tick arises informing it's correct to the user. A nice time line is provided to show the deadline seconds graphically. The line ends correctly after 20 seconds, it moves from left to right in such a way that it starts at 1 and ends at 20. The fourth card holds the aptitude questions for practice, on entering this card the user will be able to see the top 5 topics of aptitude questions with a small description about each topic on their respective cards. The user can choose the topic of their interest the test rules are displayed describing the rules of the test. For aptitude we require atleast a minute to solve one question. We have encoded one minute as the time limit

per question. The aim of doing this is for the students to understand their speed of problem solving. The speed and accuracy of solving each question matters in the field of aptitude. If the student takes up the test and gives a wrong answer the correct answer with solution will be displayed. If the student does not answer within the stipulated time then the answer will automatically be displayed at the end of 1 min (60 seconds). If the student chooses the right answer the solution to the same will be displayed. You might think why to display the solution when the chosen answer is correct. There is a reason to that, There maybe situations where the student by luck can choose the right answer or might have thought of the solution in a different angle which might not work for all such types of problems. We provide the solution for the student to cross check their logical match. The student is also given an opportunity to work on the same set of questions on loop with the help of retest option. So overall these requirements are beneficial for last minute preparations. Hence the students can use our website even a day before the interview call.

The placements view is created for the placement department. Only the placement head can access the placement webpage. The details about the students and their status of placement will be updated through database by the placement coordinator team. The placement head will be worried only about the status of the placements so only that is made visible to the placement head, hence we abstract the database entry view. This view would help the placement head to know the status of student’s progress and can take required action to help the unplaced student to get placed. Hence hire hub serves efficiently to both the placement head and also the students appearing for placements.

3.1 Hardware Requirements

➢ Processor: Intel Core i5

➢ Memory: 16 KB

➢ Hard Disk: 24 K

3.2 Software Requirements

➢ Operating System: Windows 10

➢ Back – end: JavaScript , database(Phpmyadmin)

➢ Front – end: HTML and CSS

➢ Visual Studio Code (or) Notepad

➢ Xampp Server

CHAPTER 4

4 System Design

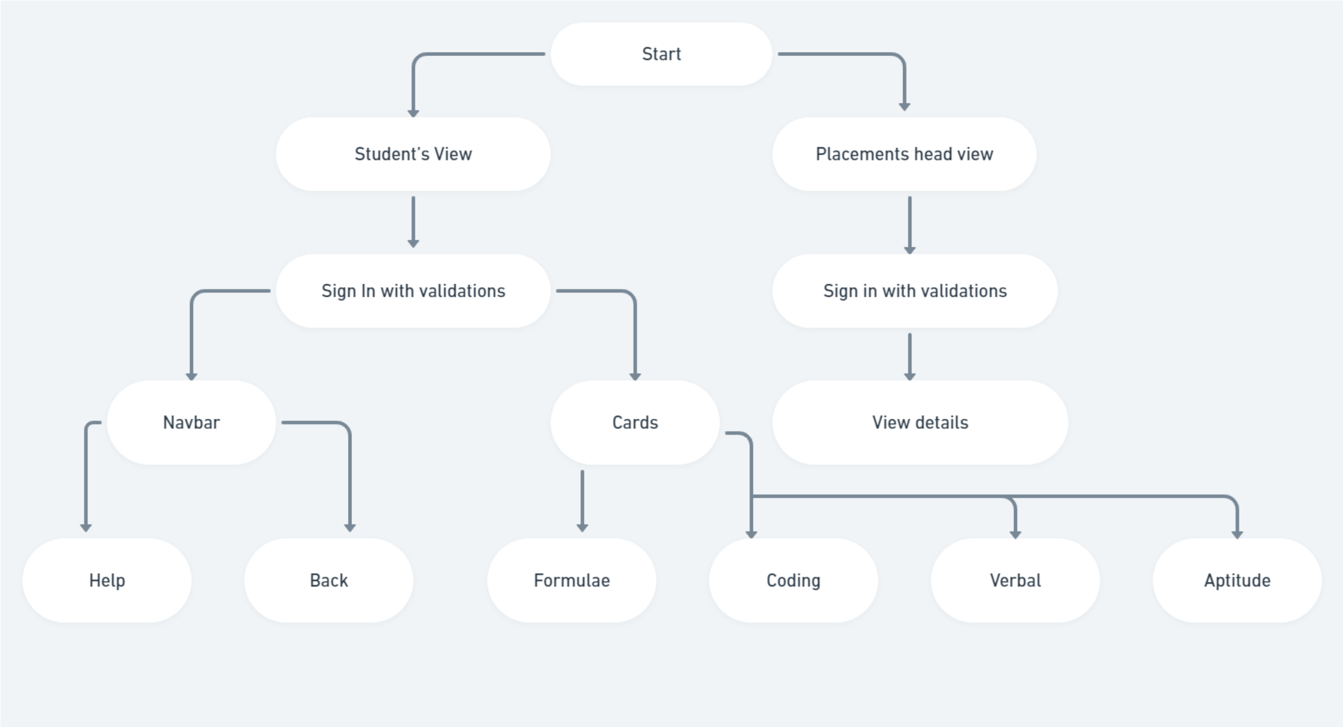


Fig-1.1 System Design

As discussed earlier we have two views on our website. In the students view we will have a sign in form if and only if the person owns an account on hire hub he/she will be allowed to enter the site. After entering the site the student can know more about hire hub by reading the help page directed by navbar. There are 4 cards on the page each having its own unique operation. First one displays the formulae list, second displays the coding, third displays the verbal and the last displays the aptitude. There will be multiple operations within every card. The student can choose the card of his/her interest and can start practicing. Placements head view will be authorized only to the head of placements department. He/she can view the status of placements happened so far and can also update the details based on the requirement.

4.1 Architectural Design

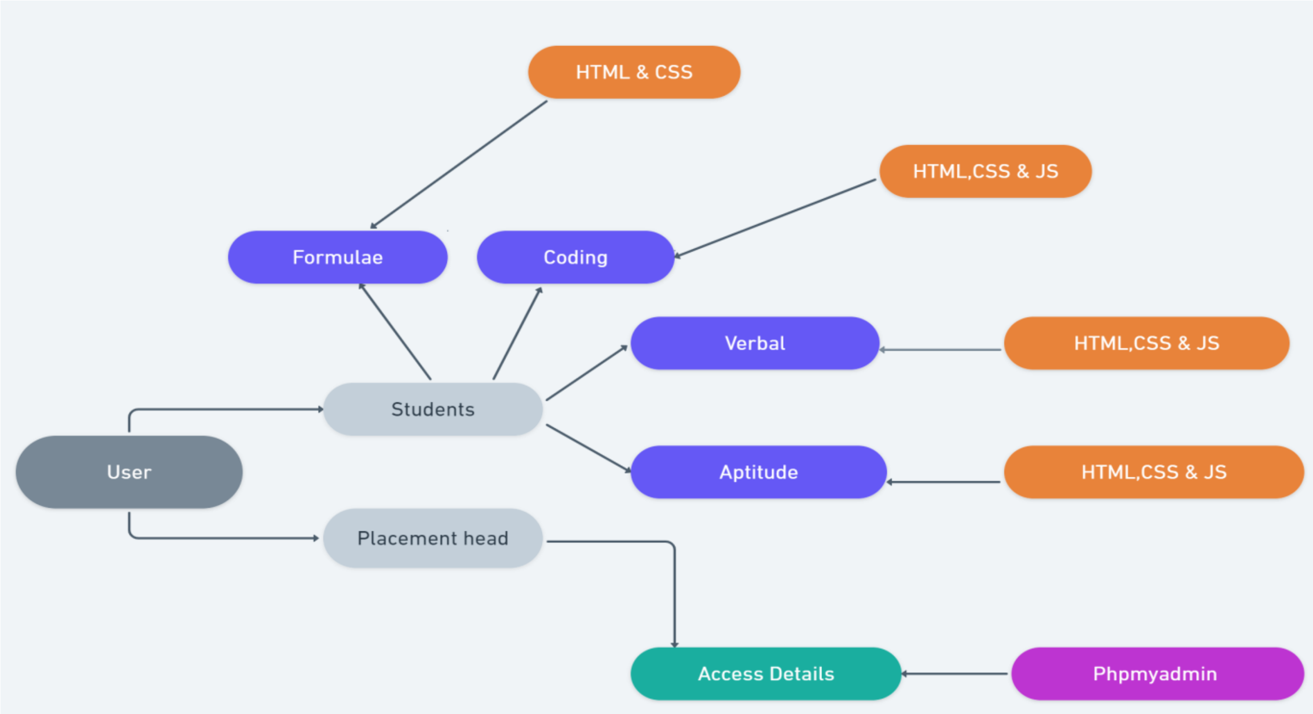


Fig-1.2 Architectural Design

We have implemented concepts of HTML, CSS, JS and phpmyadmin(database) to develop this website named hire hub. We have hardcoded formulae webpage using HTML and CSS. Coding, verbal and aptitude webpages are developed using HTML, CSS and JS. JS holds all the logic related to the respective webpages. We access the details from phpmyadmin for the placement head to view the respective details.

CHAPTER 5

5 Implementation

We have implemented two views using buttons. The student view will lead you to a sign in page the validations are properly encoded using JS. If and only if the student owns an account he will be allowed to enter into hire hub training website. After entering with valid user name and password the student will be entering the training kit webpage. Here we will have a navbar having help and back. Help will contain the information about the Hire Hub website using which the students can know the benefits of the hire hub website. We have 4 flipping cards on the middle of the webpage. First one serves the purpose of formulae done using HTML and CSS we have also included animations over the heading. Second card serves the purpose of coding entering which the student can explore three different levels of coding. The solutions to each question is encoded using JS. Third and fourth card serves the purpose of verbal and aptitude tests respectively. Both are encoded using HTML,CSS and JS. Here we have included time constraints and answer displays. Time for verbal is encoded to be 20 seconds and time for aptitude is encoded to be 60 seconds. As soon as u enter one of these section named verbal or aptitude it will provide you 5 different topic options, you can choose a topic according to your interest and take up the test. The test rules are displayed using a modal agreeing to which you can take part in the test. The option of retake test is also made available to the students taking up the test. The placement button will lead you to a sign in page and this is accessable only by the placement head. He/she will be able to view and update the placement details of any student of the current batch. This will help the placement head to know the status and progress of the placements of each student. The decision of changing the training methodology depends on how good or how bad the placements status remains. This aims to provide 100% placement to all the students.

5.1 System Modules

We have many modules where each is designed to deal with a unique operation. Here we shall discuss the importance of each module. Every html file is separately associated with CSS

- HH_student.html- This page aims to provide the framework of the students profile page. This page is accessable only after signing in with valid user name and password. It contains the navbar directing to the help page .It also contains four cards for respective operations. First for formulae, second for coding, third for verbal and fourth for aptitude. The flip cards are designed to give a professional look to the website.
- Formulae.html- This page aims to design the formulae page. Animation styling is given to the heading with timeout. The formula of each topic is arranged horizontally in a table format enclosed within the border. A back button is provided to direct you back to the previous webpage.
- Coding.html- This page is designed to provide three cards containing three different coding levels. There are easy, medium and hard levels each containing respective type of questions. Each question is embedded with clear solution, there is a toggle window to view the solution. There is a back button to re-direct to the previous webpage.
- Verbal.html- This page is designed to provide 5 cards framework. Each card contains a topic name. There is a small description for each topic. There are separate html and js files for each topic. In js we have encoded the test attributes. The time limitations, time line, questions and answers are encoded in js. The time line starts running from 1 and ends at 20 hence provides a nice professional deadline.
- Aptitude.html- This page is designed to provide 5 cards framework. Each card contains a topic name. There is a small description for each topic. There are separate html and js files for each topic. In js we have encoded the test attributes. The time limitations, time line,

questions and answers are encoded in js. The time line starts running from 1 and ends at 60 hence provides a nice professional deadline.

- SignIn.html- This page is designed to provide authorized access to the students owning an account on hire hub. The validations are hardcoded using JS. We get clear prompts when the user validations are not satisfied. Hence it helps the users to understand the error involved with the access denial.
- pcSignIn.html- This page is designed to provide authorized access to the placement head owning an account on hire hub. The validations are hardcoded using JS. We get clear prompts when the user validations are not satisfied. Hence it helps the users to understand the error involved with the access denial. This sign in provides access only to the placement head.

5.2 Algorithm

```
Step 1:  Start
Step 2:  Choose the user type- placement_head or student.
Step 3:  If student you will be provided with 4 cards each containing a segment of training.
Step 4:  If you choose formulae a page containing formulae opens.
Step 5:  If you choose coding the coding page with three levels of coding opens.
Step 6:  If you choose verbal the 5 cards holding 5 verbal topics opens.
Step 7:  Under verbal you can choose your interested topic and take up test.
Step 8:  If you choose aptitude the 5 cards holding 5 aptitude topics opens.
Step 9:  Under aptitude you can choose your interested topic and take up test.
Step 10: If placement-head you will be given access to view and update the status of any student.
```

Step 11: Stop

5.3 Code

HH.html

```
<!DOCTYPE html>
<html>
<head>
     <title> Hire Hub </title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
     <link  href=https://fonts.googleapis.com/css2?family=Noto+Sans&display=swap
     rel="stylesheet">                                                    <link
     rel="stylesheet"href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/boo
     tstrap.min.css" >
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
      rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="hh.css">
</head>
<style>
       .wrap {
                    height: 100%;
                    display: flex;
                    align-items: center;
                    justify-content: center;
                  }
      .button {
```

```
                    min-width: 500px;
                    min-height: 80px;
                    font-family: 'Nunito', sans-serif;
                    font-size: 22px;

                      text-transform: uppercase;
                      letter-spacing: 1.3px;
                      font-weight: 700;
                      color: #313133;
                      background: #4FD1C5;
                       background:  linear-gradient(90deg,  rgba(129,230,217,1)  0%,  rgba(79,209,197,1)
                      100%);
                     border: none;
                     border-radius: 1000px;
                     box-shadow: 12px 12px 24px lightsteelblue;
                     transition: all 0.3s ease-in-out 0s;
                     cursor: pointer;
                     outline: none;
                     position: relative;
                     padding: 10px;
                }
.button:hover, .button:focus {
                     color: #313133;
                     transform: translateY(-6px);
                                           }
button::after {
                    width: 30px; height: 30px;
                    border-radius: 100%;
                    border: 6px solid #00FFCB;
                   }
button::before{
```

```
                    content: ";
                    border-radius: 1000px;
                    min-width: calc(300px + 12px);
                    min-height: calc(60px + 12px);
                  border: 6px solid #00FFCB;
                  box-shadow: 0 0 60px rgba(0,255,203,.64);
                  position: absolute;
                  top: 50%;
                  left: 50%;
                  transform: translate(-50%, -50%);
                  opacity: 0;
                  transition: all .3s ease-in-out 0s;
                }
</style>
<body>
      <section id= "container">
            <div style="background-color:lightskyblue">H</div>
            <div style="background-color:lightseagreen">I</div>
            <div style="background-color:lightskyblue;height:100px;">R</div>
            <div style="background-color:lightseagreen">E</div>
            <div style="background-color:lightskyblue;height:100px;">H</div>
            <div style="background-color:lightseagreen">U</div>
            <div style="background-color:lightskyblue">B</div>
    </section>
    <br><br> <br><br><br><br><br>
    <p>
        <div class="wrap">
            <a href="signin.html">
                   <button class="button" >Student</button>
              </a>
         </div>
        </p>
        <br>
        <p>
            <div class="wrap">
             <a href="pcSignIn.html">
                    <button class="button">Placement Coordinator</button>
             </a>
           </div>
        </p>
   </body>
   </html>
```

HH_student.html

```
<html>
<head>
       <title>student profile</title>
       <link rel="stylesheet" type="text/css" href="hh_student.css">
       <link rel="preconnect" href="https://fonts.googleapis.com">
       <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
        <linkhref="https://fonts.googleapis.com/css2?family=Noto+Sans&display=swap"
        rel="stylesheet">
        <linkrel="stylesheet"href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.mi
        n.css" >
      <link href="https://fonts.googleapis.com/icon?family=Material+Icons”  rel="stylesheet">
</head>
<body>
       <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container-fluid">
```

```
   <!-- Brand and toggle get grouped for better mobile display -->
   <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
        target="#bs-example-navbar-collapse-1" aria-expanded="false">
              <span class="sr-only">Toggle navigation</span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
        </button>
      <a class="navbar-brand" href="#"><div><div class="glyphicon glyphicon-eye-
      open"></div>Happy Learning</div></a>
      </div>
      <ul class="nav navbar-nav navbar-right">
         <li><a href="help.html">Help</a></li>
         <li><a href="hh.html">Log Out</a></li>
         </li>
      </ul>
    </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
  </nav>
<br><br><br><br><br>
<div class="row">
  <div class="column">
      <div class="flip-card">
        <div class="flip-card-inner">
          <div class="flip-card-front">
            <h1>Formulae</h1>
          </div>
          <div class="flip-card-back">
            <h1>Formulae Topic Wise</h1>
```

```
                  <br>
                 <ahref="Formulae/formulae.html"><buttonclass="b1"><span>Go!</span></button>
                 </a>
                </div>
            </div>
        </div>
        </div>
        <div class="column">
        <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <h1>Coding</h1>
                </div>
                <div class="flip-card-back">
                    <h1>Let's Test Your Coding Skills</h1>
                    <br>
                    <a href="Coding/coding.html">
                         <button class="b2"><span>Go!</span></button>
                    </a>
                </div>
            </div>
        </div>
    </div>
    </div>
<div class="row">
    <div class="column">
        <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <h1>Verbal</h1>
                </div>
```

```
                    <div class="flip-card-back">
                        <h1>Take Up Verbal Test</h1>
                        <br>
                        <ahref="Verbal/verbal.html"><buttonclass="b3"><span>Go!</span></button></a>
                    </div>
                </div>
            </div>
        </div>
        <div class="column">
            <div class="flip-card">
                <div class="flip-card-inner">
                    <div class="flip-card-front">
                        <h1>Aptitude</h1>
                    </div>
                    <div class="flip-card-back">
                        <h1>Take Up Aptitude Test</h1>
                        <br>
                        <a href="aptitude/aptitude.html">
                        <button class="b4"><span>Go!</span></button>
                        </a>
                    </div>
                </div>
            </div>
        </div>
        </div>
</body>
</html>
```

Coding.html

```
<html>
<head>
    <title>Let's Code</title>
    <link rel="stylesheet" type="text/css" href="coding.css">

<linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.cs"
    integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
</head>
<body>
    <div class="center">
        <h1>Choose Your Level Of Coding</h1>
    </div>
    <div id="codingLevelContainer" class="d-flex">
        <div id="easyContainer">
            <h2>EASY</h2>
            <p><span>Copy-and-Paste was programmed by programmers for programmers
actually</span> ,Wait did you just smile? Yes, this coding is just as simple as that smile on your
face right now... so what are you waiting for let's get started!</p>
            <a href="Easy/easy.html"><button class="easy">GO!</button></a>
        </div>
        <div id="mediumContainer">
            <h2>MEDIUM</h2>
            <p><span>while(!(succeed=try()));</span> The only thing that matters is your effort to
solve the problem. Solution is always the product of effort.Keep trying without having much
concern on the solution being right.I am sure you can do it try me!</p>
            <a href="Medium/medium.html"><button class="medium">GO!</button></a>
        </div>
        <div id="hardContainer">
```

```
            <h2>HARD</h2>
            <p><span>Remember no code has zero defects</span> Solutions might differ all that matters is your approach towards a real time problem. Its not too hard for you to give up... give it a try and find the bugs with your thinking process!!!</p>
            <a href="Hard/hard.html"><button class="hard">GO!</button></a>
        </div>
    </div>
    <div id="returnContainer" class="d-flex justify-content-center">
        <a href="../hh_student.html"><button class="return">Back</button></a>
    </div>
</body>
</html>
```

Verbal.html

```
<!DOCTYPE html>
<html>
<title>Verbal</title>
<head>
     <meta charset="UTF-8">
     <meta name="viewport" content=" width=device-width, initial-scale=1.0">
     <linkrel="stylesheet"type="text/css"href="verbal.css">
     <linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.
     css"
     integrity="sha384-
     Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
     crossorigin="anonymous">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <title>Verbal</title>
</head>
```

```
<body>
  <h1 class="title">
      <span class="title-word title-word-1">Verbal</span>
      <span class="title-word title-word-2">Test</span>
  </h1>
  <h3> Select one of the following topics</h3>
  <br><br>
  <div id="codingLevelContainer" class="d-flex">
    <div id="Container1">
            <p><span>Idioms and phrases,</span> collection of words which are not easily
            understandable, It also preserves a special meaning. In simple words its just like the
            normal sarcasms we use. come on lets explore a few of them!</p>
        <ahref="Idioms/idioms.html"><buttonclass="verbalOptionsButton">GO!</button></a>
    </div>
    <div id="Container2">
             <p><span>Synonyms,</span> Words with similar meaning. lets brush up a little
            before we get started...beat->defeat, tender->soft, easy->simple, hot->burning, hard-
            >challenging come let's explore more of it</p>
           <ahref="Synonyms/synonyms.html"><buttonclass="verbalOptionsButton">GO!</bu
           tton></a>
    </div>
    <div id="Container3">
              <p><span>Antonyms,</span> Words with opposite meaning. lets brush up a
             little before we get started...joy->grief, borrow->lend, absent->present, dry->wet,
             after->before come let's explore more of it </p>
             <ahref="Antonyms/antonyms.html"><buttonclass="verbalOptionsButton">GO!</
             button></a>
    </div>
    <div id="Container4">
```

```
                      <p><span>Sentence Correction,</span> The purpose of this exercise is to help
                      us build our grammar. We tend to make alot of errors with tenses so here is an
                      helping hand to fix our grammarly bugs. </p>
                      <ahref="SentenceCorrection/sentenceCorrection.html"><button
                      class="verbalOptionsButton">GO!</button></a>
         </div>
        <div id="Container5">
                      <p><span>Sentence Formation,</span><br> This is to help us prioritize words.
                      The understanding depends on the arrangement of words. Without arrangement
                      words are totally meaningless demo -chocolates,I,eat,tend,midnight,at,to come
                      on lets fix this. </p>
                      <ahref="SentenceFormation/sentenceFormation.html"><button
                      class="verbalOptionsButton">GO!</button></a>
         </div>
       </div>
       <div id="returnContainer" >
                <a href="../hh_student.html"><button class="return">Back</button></a>
       </div>
    </div>
</body>
</html>
```

AptitudeTest.js

```
const start_btn = document.querySelector(".start_btn button");
const info_box = document.querySelector(".info_box");
const exit_btn = info_box.querySelector(".buttons .quit");
const continue_btn = info_box.querySelector(".buttons .restart");
const quiz_box = document.querySelector(".quiz_box");
const result_box = document.querySelector(".result_box");
const option_list = document.querySelector(".option_list");
const time_line = document.querySelector("header .time_line");
const timeText = document.querySelector(".timer .time_left_txt");
const timeCount = document.querySelector(".timer .timer_sec");
const solution = document.querySelector(".solution");

// if startQuiz button clicked
start_btn.onclick = ()=>{
   info_box.classList.add("activeInfo"); //show info box
}

// if exitQuiz button clicked
exit_btn.onclick = ()=>{
   info_box.classList.remove("activeInfo"); //hide info box
}

// if continueQuiz button clicked
continue_btn.onclick = ()=>{
   info_box.classList.remove("activeInfo"); //hide info box
   quiz_box.classList.add("activeQuiz"); //show quiz box
   showQuetions(0); //calling showQestions function
   queCounter(1); //passing 1 parameter to queCounter
   startTimer(60); //calling startTimer function
   startTimerLine(0); //calling startTimerLine function
}

let timeValue =  60;
let que_count = 0;
let que_numb = 1;
let userScore = 0;
let counter;
let counterLine;
let widthValue = 0;

const restart_quiz = result_box.querySelector(".buttons .restart");
const quit_quiz = result_box.querySelector(".buttons .quit");

// if restartQuiz button clicked
restart_quiz.onclick = ()=>{
    quiz_box.classList.add("activeQuiz"); //show quiz box
    result_box.classList.remove("activeResult"); //hide result box
    timeValue = 60;
    que_count = 0;
    que_numb = 1;
    userScore = 0;
    widthValue = 0;
    showQuetions(que_count); //calling showQestions function
    queCounter(que_numb); //passing que_numb value to queCounter
    clearInterval(counter); //clear counter
    clearInterval(counterLine); //clear counterLine
    startTimer(timeValue); //calling startTimer function
    startTimerLine(widthValue); //calling startTimerLine function
    timeText.textContent = "Time Left"; //change the text of timeText to Time Left
    next_btn.classList.remove("show"); //hide the next button
}

// if quitQuiz button clicked
quit_quiz.onclick = ()=>{
    window.location.reload(); //reload the current window
}
```

```
const next_btn = document.querySelector("footer .next_btn");
const bottom_ques_counter = document.querySelector("footer .total_que");

// if Next Que button clicked
next_btn.onclick = ()=>{
    solution.innerHTML = "";
    if(que_count < questions.length - 1){ //if question count is less than total question length
        que_count++; //increment the que_count value
        que_numb++; //increment the que_numb value
        showQuetions(que_count); //calling showQestions function
        queCounter(que_numb); //passing que_numb value to queCounter
        clearInterval(counter); //clear counter
        clearInterval(counterLine); //clear counterLine
        startTimer(timeValue); //calling startTimer function
        startTimerLine(widthValue); //calling startTimerLine function
        timeText.textContent = "Time Left"; //change the timeText to Time Left
        next_btn.classList.remove("show"); //hide the next button
    }else{
        clearInterval(counter); //clear counter
        clearInterval(counterLine); //clear counterLine
        showResult(); //calling showResult function
    }
}

// getting questions and options from array
function showQuetions(index){
    const que_text = document.querySelector(".que_text");
//creating a new span and div tag for question and option and passing the value using array index
    let que_tag = '<span>'+ questions[index].numb + ". " + questions[index].question +'</span>';
    let option_tag = '<div class="option"><span>'+ questions[index].options[0] +'</span></div>'
    + '<div class="option"><span>'+ questions[index].options[1] +'</span></div>'
    + '<div class="option"><span>'+ questions[index].options[2] +'</span></div>'
    + '<div class="option"><span>'+ questions[index].options[3] +'</span></div>';
    que_text.innerHTML = que_tag; //adding new span tag inside que_tag
    option_list.innerHTML = option_tag; //adding new div tag inside option_tag

    const option = option_list.querySelectorAll(".option");

    // set onclick attribute to all available options
    for(i=0; i < option.length; i++){
        option[i].setAttribute("onclick", "optionSelected(this)");
    }
}
// creating the new div tags which for icons
let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';
let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';

//if user clicked on option
function optionSelected(answer){
    clearInterval(counter); //clear counter
    clearInterval(counterLine); //clear counterLine
    let userAns = answer.textContent; //getting user selected option
    let correcAns = questions[que_count].answer; //getting correct answer from array
    const allOptions = option_list.children.length; //getting all option items

    if(userAns == correcAns){ //if user selected option is equal to array's correct answer
        userScore += 1; //upgrading score value with 1
        answer.classList.add("correct"); //adding green color to correct selected option
        answer.insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to correct selected option
        console.log("Correct Answer");
        console.log("Your correct answers = " + userScore);
    }else{
        answer.classList.add("incorrect"); //adding red color to correct selected option
        answer.insertAdjacentHTML("beforeend", crossIconTag); //adding cross icon to correct selected option
        console.log("Wrong Answer");

        for(i=0; i < allOptions; i++){
            if(option_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer
                option_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option
                option_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option
                console.log("Auto selected correct answer.");
            }
        }
    }
    for(i=0; i < allOptions; i++){
        option_list.children[i].classList.add("disabled"); //once user select an option then disabled all options
    }
    next_btn.classList.add("show"); //show the next button if user selected any option

    showSolution();
}

function showResult(){
    info_box.classList.remove("activeInfo"); //hide info box
    quiz_box.classList.remove("activeQuiz"); //hide quiz box
    result_box.classList.add("activeResult"); //show result box
    const scoreText = result_box.querySelector(".score_text");
    if (userScore > 3){ // if user scored more than 3
        //creating a new span tag and passing the user score number and total question number
        let scoreTag = '<span>and congrats! , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;  //adding new span tag inside score_Text
    }
    else if(userScore > 1){ // if user scored more than 1
        let scoreTag = '<span>and nice , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
    else{ // if user scored less than 1
        let scoreTag = '<span>and sorry , You got only <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
}
function startTimer(time){
    counter = setInterval(timer, 1000);
    function timer(){
        timeCount.textContent = time; //changing the value of timeCount with time value
        time--; //decrement the time value
        if(time < 9){ //if timer is less than 9
            let addZero = timeCount.textContent;
            timeCount.textContent = "0" + addZero; //add a 0 before time value
        }
        if(time < 0){ //if timer is less than 0
            clearInterval(counter); //clear counter
            timeText.textContent = "Time Off"; //change the time text to time off
            const allOptions = option_list.children.length; //getting all option items
            let correcAns = questions[que_count].answer; //getting correct answer from array
            for(i=0; i < allOptions; i++){
                if(option_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer
                    option_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option
                    option_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option
                    console.log("Time Off: Auto selected correct answer.");
                }
            }
            for(i=0; i < allOptions; i++){
                option_list.children[i].classList.add("disabled"); //once user select an option then disabled all options
            }
            next_btn.classList.add("show"); //show the next button if user selected any option

            showSolution();
        }
    }
}

function startTimerLine(time){
    counterLine = setInterval(timer, 0);
    function timer(){
        time += 0.048; //upgrading time value with 1
        time_line.style.width = time + "px"; //increasing width of time_line with px by time value
        if(time > 710){ //if time value is greater than 549
            clearInterval(counterLine); //clear counterLine
        }
    }
}
```

```
function queCounter(index){
    //creating a new span tag and passing the question number and total question
    let totalQueCounTag = '<span><p>'+ index +'</p> of <p>'+ questions.length +'</p> Questions</span>';
    bottom_ques_counter.innerHTML = totalQueCounTag;  //adding new span tag inside bottom_ques_counter
}

function showSolution() {
    if(questions[que_count].solution != "" || questions[que_count].solution == undefined) {
        let solutionText = "<h4>Solution<h4><br>" + questions[que_count].solution;
        solution.innerHTML = solutionText;
    }
}
```

CHAPTER 6

6 Results

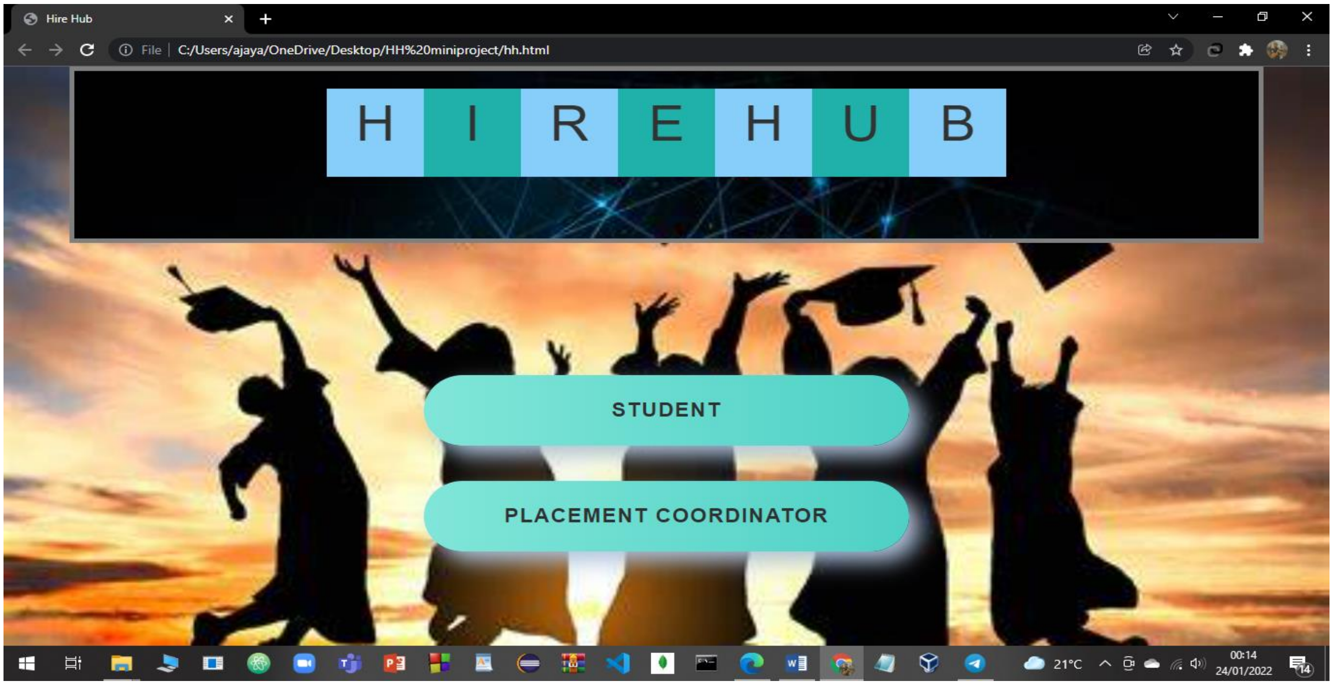


Fig-1.3 Hire Hub HomePage

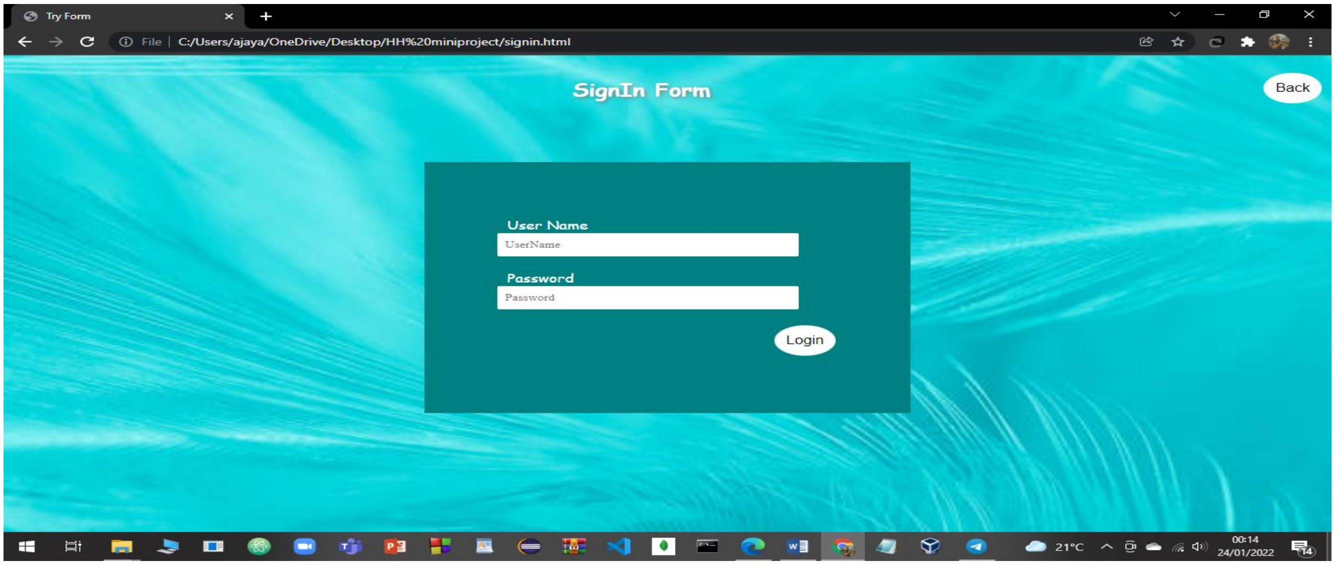


Fig-1.4 SignIn For Student

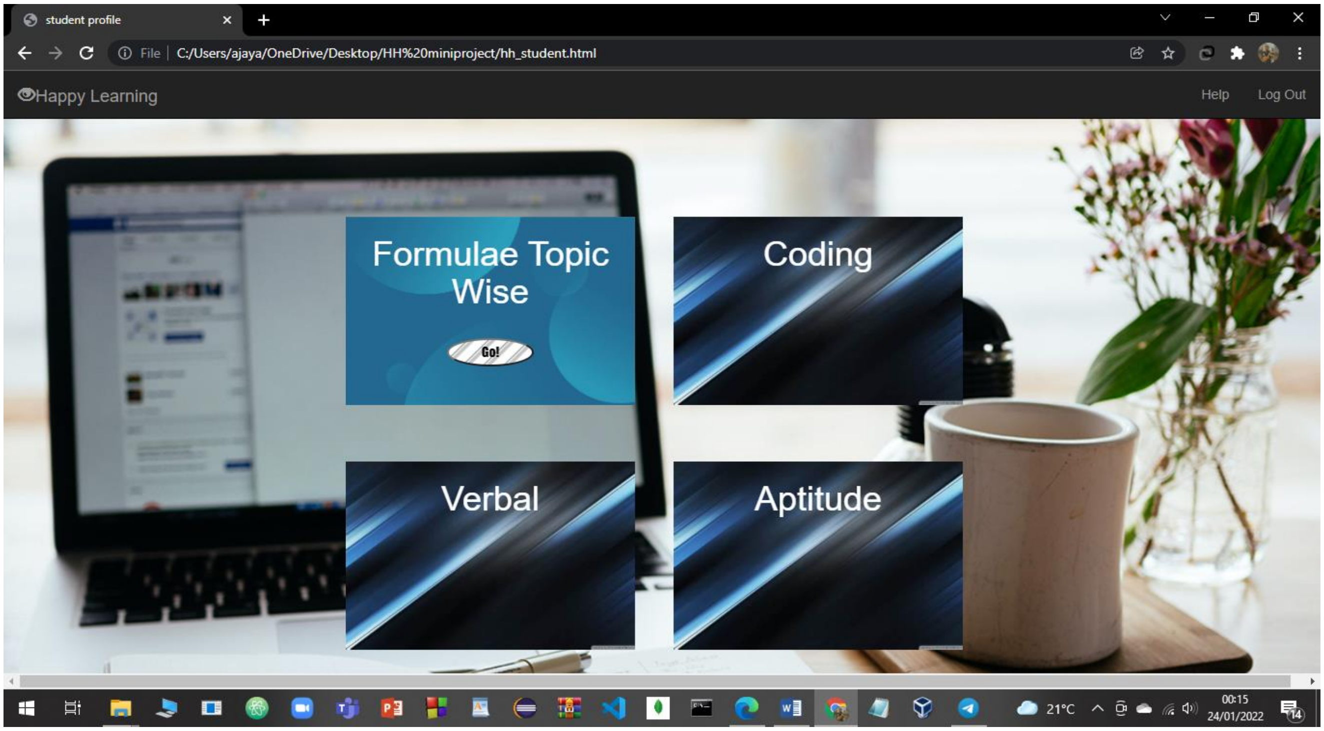
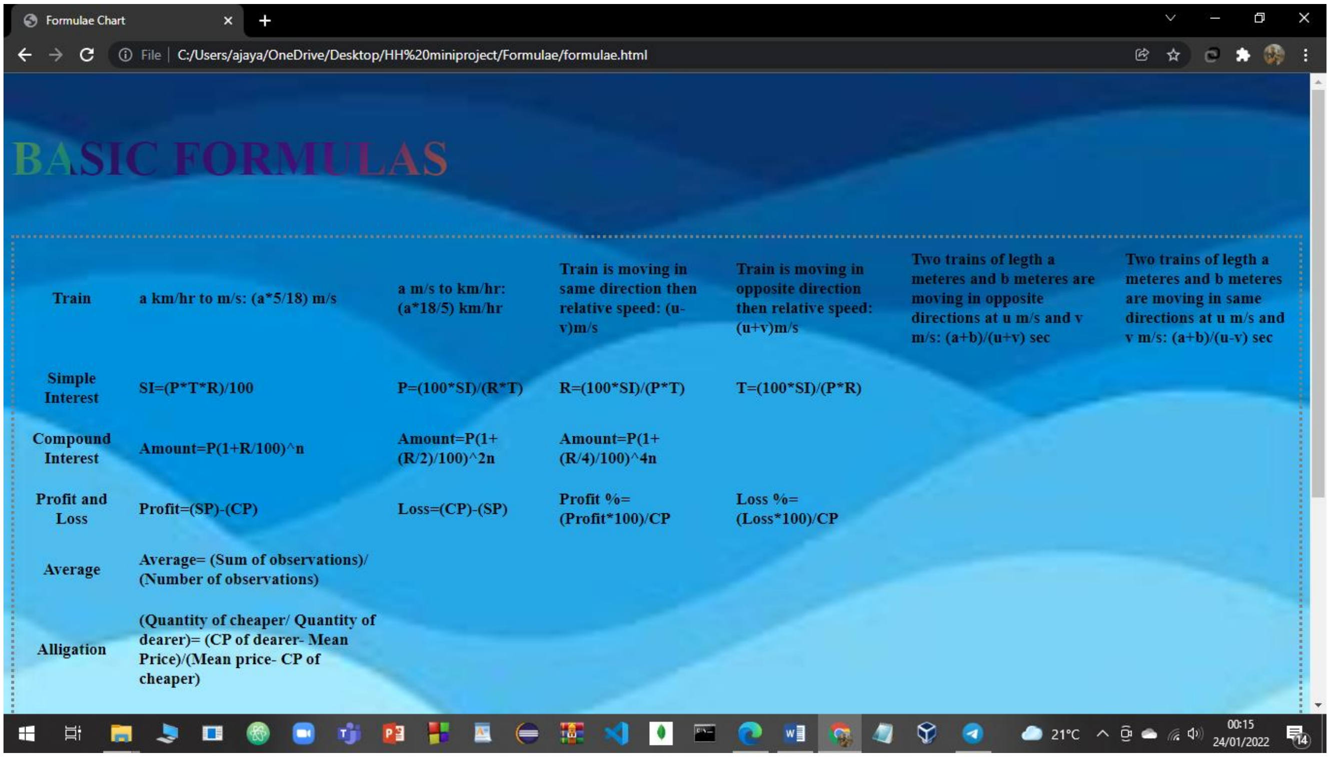


Fig-1.5 Formulae flip card


Formulae Chart
C:/Users/ajaya/OneDrive/Desktop/HH%20miniproject/Formulae/formulae.html
21°C
00:15
24/01/2022

BASIC FORMULAS

Train	a km/hr to m/s: (a*5/18) m/s	a m/s to km/hr: (a*18/5) km/hr	Train is moving in same direction then relative speed: (u-v)m/s	Train is moving in opposite direction then relative speed: (u+v)m/s	Two trains of legth a meteres and b meteres are moving in opposite directions at u m/s and v m/s: (a+b)/(u+v) sec	Two trains of legth a meteres and b meteres are moving in same directions at u m/s and v m/s: (a+b)/(u-v) sec
Simple Interest	SI=(P*T*R)/100	P=(100*SI)/(R*T)	R=(100*SI)/(P*T)	T=(100*SI)/(P*R)		
Compound Interest	Amount=P(1+R/100)^n	Amount=P(1+(R/2)/100)^2n	Amount=P(1+(R/4)/100)^4n			
Profit and Loss	Profit=(SP)-(CP)	Loss=(CP)-(SP)	Profit %=(Profit*100)/CP	Loss %=(Loss*100)/CP		
Average	Average= (Sum of observations)/(Number of observations)					
Alligation	(Quantity of cheaper/ Quantity of dearer)= (CP of dearer- Mean Price)/(Mean price- CP of cheaper)					

Fig-1.6 Basic formulas

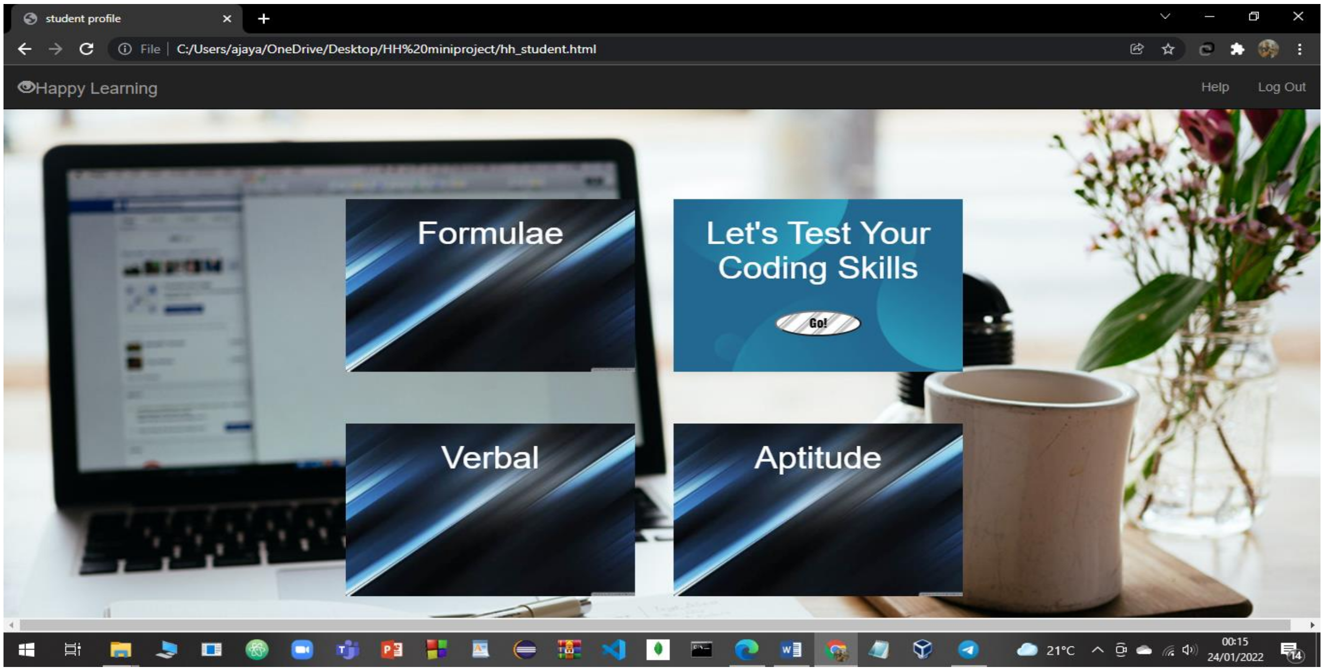
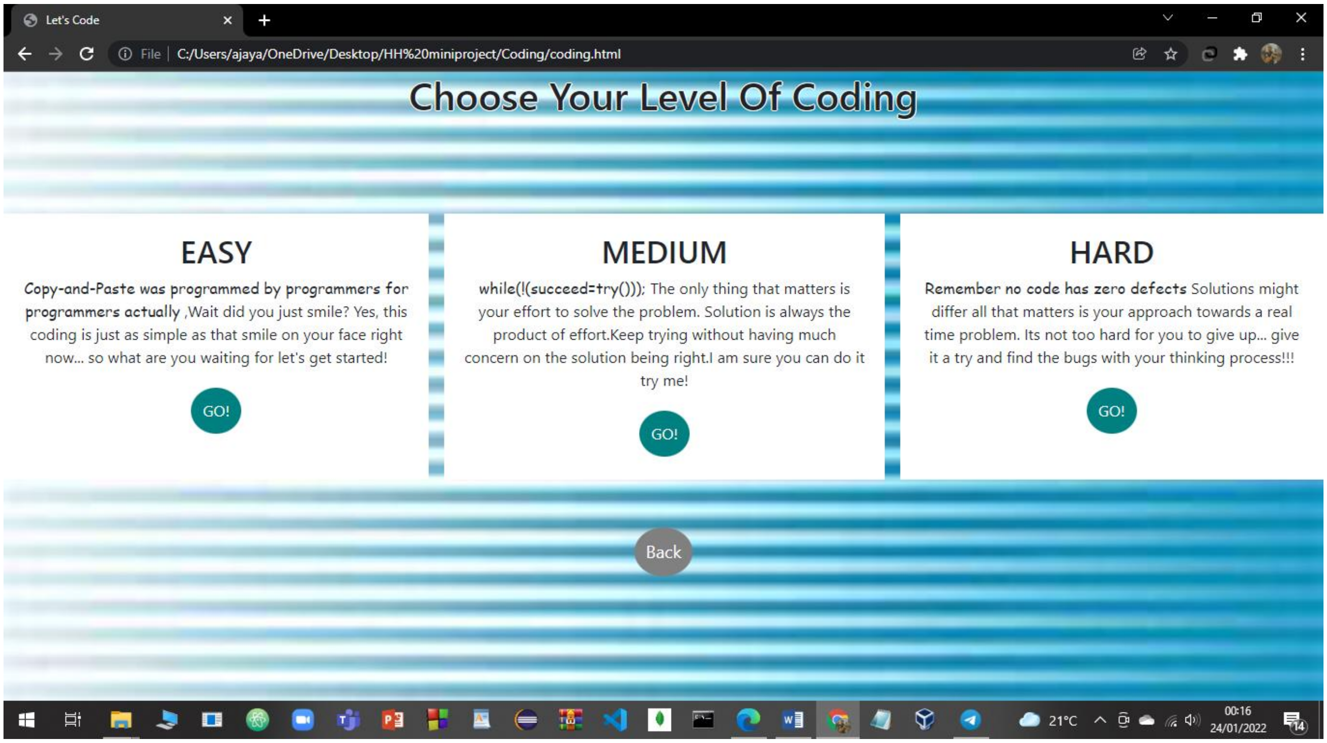

student profile
C:/Users/ajaya/OneDrive/Desktop/HH%20miniproject/hh_student.html
Happy Learning
Help
Log Out
Formulae
Let's Test Your Coding Skills
Go!
Verbal
Aptitude
21°C
00:15
24/01/2022

Fig-1.7 Coding Flip Card


Let's Code
C:/Users/ajaya/OneDrive/Desktop/HH%20miniproject/Coding/coding.html

Choose Your Level Of Coding

EASY

Copy-and-Paste was programmed by programmers for programmers actually ,Wait did you just smile? Yes, this coding is just as simple as that smile on your face right now... so what are you waiting for let's get started!

MEDIUM

while(!(succeed=try())); The only thing that matters is your effort to solve the problem. Solution is always the product of effort.Keep trying without having much concern on the solution being right.I am sure you can do it try me!

HARD

Remember no code has zero defects Solutions might differ all that matters is your approach towards a real time problem. Its not too hard for you to give up... give it a try and find the bugs with your thinking process!!!

Fig-1.8 Level Of Coding

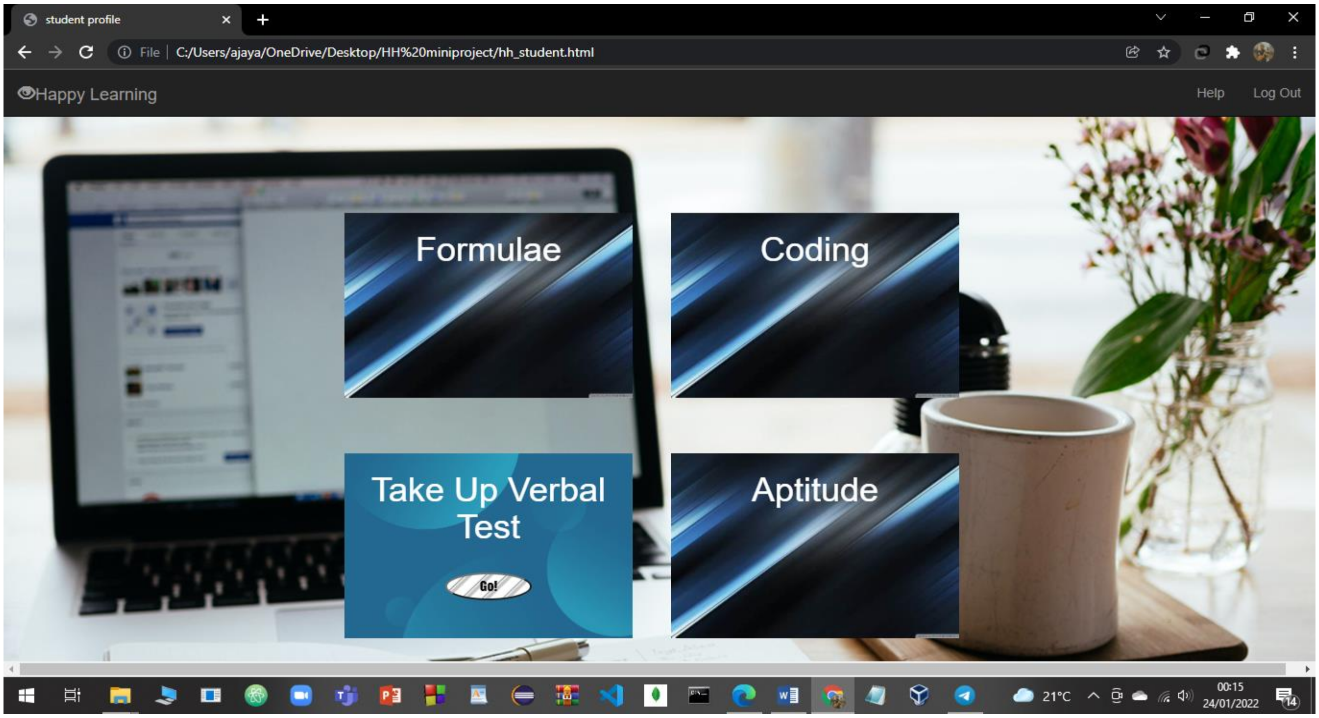
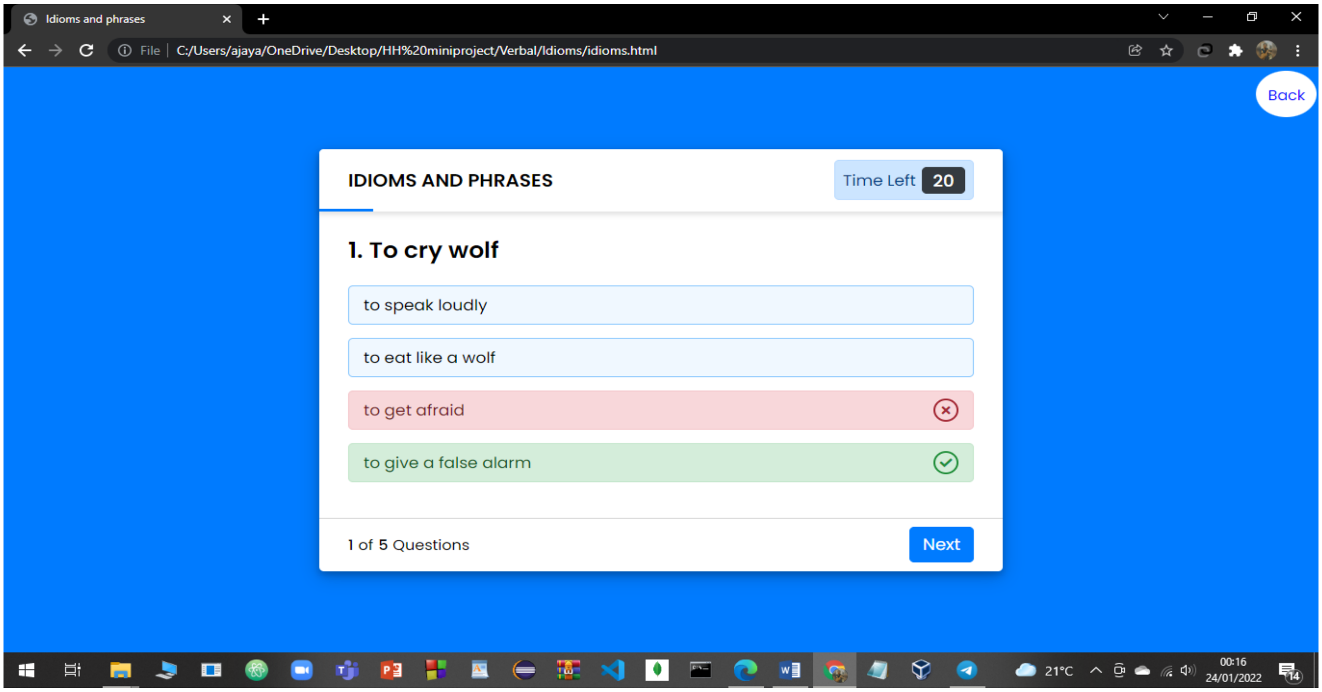


Fig-1.9 Verbal Flip Card


Idioms and phrases
C:/Users/ajaya/OneDrive/Desktop/HH%20miniproject/Verbal/Idioms/idioms.html
Back
Time Left 20
Next
21°C
00:16
24/01/2022

IDIOMS AND PHRASES

1. To cry wolf

- to speak loudly
- to eat like a wolf
- to get afraid ⊗
- to give a false alarm ✓

1 of 5 Questions

Fig-2 Idioms And Phrases

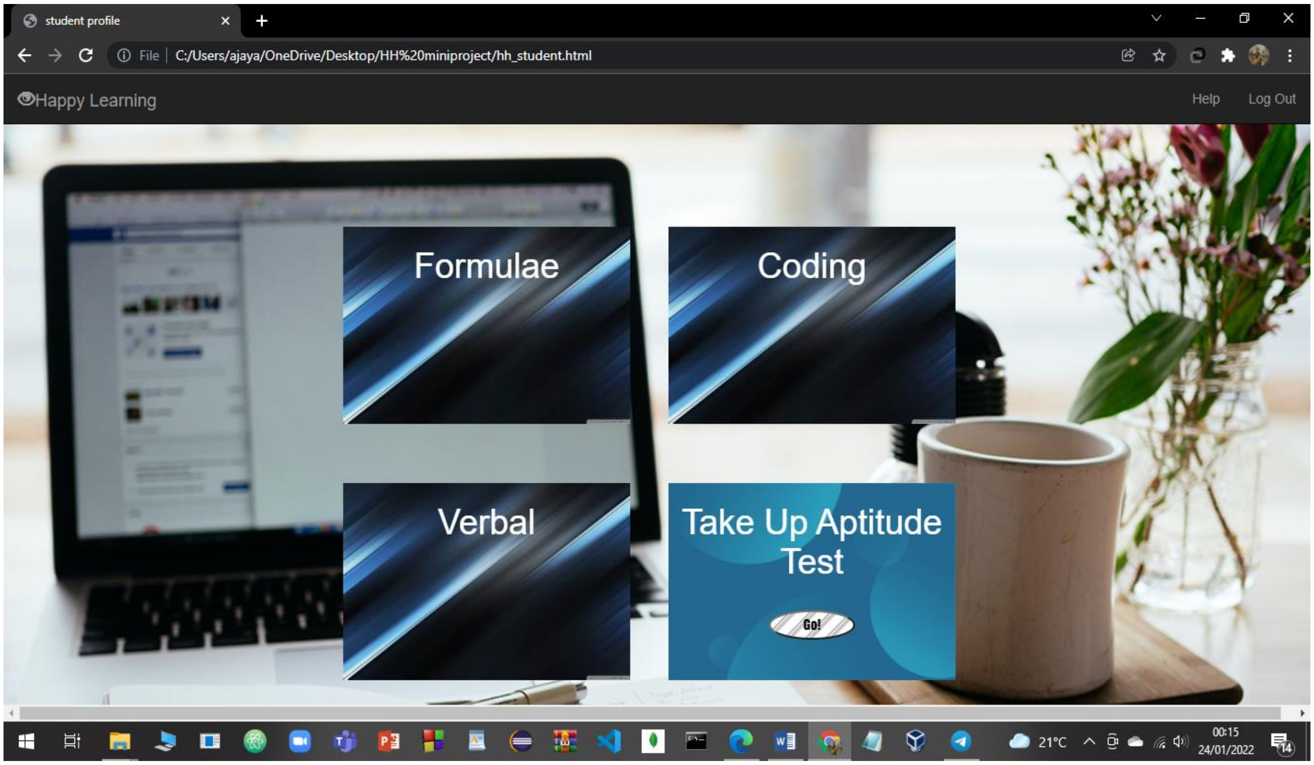
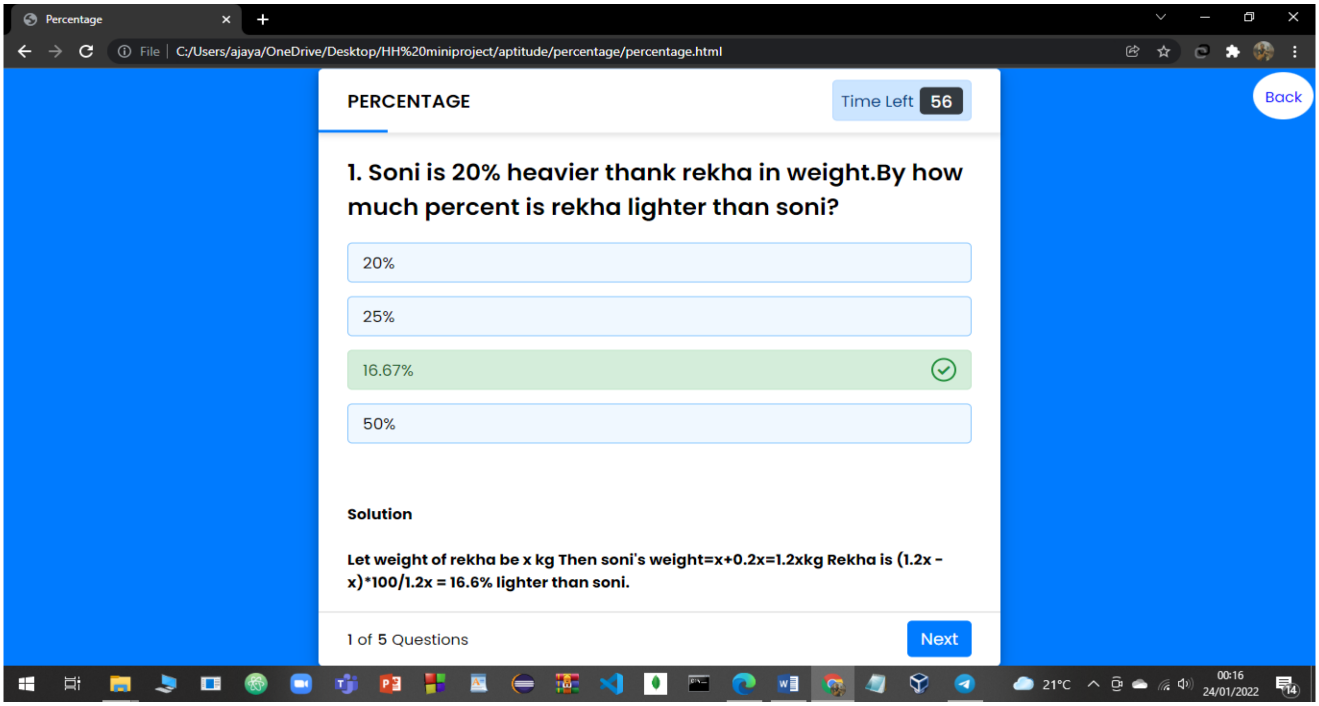


Fig-2.1 Aptitude Flip Card


Percentage
File | C:/Users/ajaya/OneDrive/Desktop/HH%20miniproject/aptitude/percentage/percentage.html
Time Left 56
Back
Next
21°C
00:16
24/01/2022

PERCENTAGE

1. Soni is 20% heavier thank rekha in weight.By how much percent is rekha lighter than soni?

20%

25%

16.67% ✓

50%

Solution

Let weight of rekha be x kg Then soni's weight=x+0.2x=1.2xkg Rekha is (1.2x - x)*100/1.2x = 16.6% lighter than soni.

1 of 5 Questions

Fig-2.2 Percentage

CHAPTER 7

7 Conclusion and Future Enhancement

Through this project we were able to understand various corner case handling methods. We had developed a placements website with two views. One is the students view with 4 cards providing efficient preparation kit. This preparation kit can be best used by last minute preparative students and also fresher's. We provide an awareness on different type of questions with time constraints hence the student will know how much they got to work to crack the desired interview. The placement head will be able to see the updated placement status of every student on the placements view. Through this project we learnt various methods of implementation of our ideas, We learnt how to better handle our bugs, We learnt how to specify constraints for our ideas and most importantly handling corner cases was a big challenge which we enjoyed doing. We would like to thank our guide who constantly kept us going throughout the project by motivating our small small progress over reviews.

This project can be developed to a bigger scale by allowing n users to access and compete with each other at real time. This would help the students realize the level of competition and would drive them wild through the preparation. The time constraints can be handled at the end of a specified time instead of handling it for every question since some might require a little longer to solve. Weekly and monthly notifications of competitions can be mailed to the student's mail id. The progress scale can be shown on the students log in profile. An option to prepare a student's resume can be provided on the profile page of the student and an option to print the same can be provided.

7.1 References

- Bootstrap components.
- Font-family
- W3 schools
- Javatpoint
- Web Developement Bootcamp(Course)