JYOTI NIVAS COLLEGE AUTONOMOUS

Post-Graduate Centre

Bangalore – 560095

MINI PROJECT LAB

Automatic Descriptive Answer Evaluation System

Submitted By

Gudiya Kumari (20MCA12)

JYOTI NIVAS COLLEGE AUTONOMOUS

Post-Graduation Centre

Bangalore – 95

CERTIFICATE

This is to certify that ***GUDIYA KUMARI (Reg. No: 20MCA12)*** has satisfactorily completed the project in **UI for Login** as a part of the Mini project "**Automatic Descriptive Answer Evaluation System"** for the II-year (III semester) MCA course in the year 2021- 2022

Staff-in-charge **Head of Department**

Examiners:

1…………………………..

2.………………………..……

ABSTRACT

Nowadays, as we are moving towards automation there is a need for an automatic answer evaluation system. Manual evaluation is a time and energy-consuming task. When the teacher evaluates any paper manually, the quality of evaluation may vary along with the emotions of the teacher. Hence, the marks distribution can sometimes be inappropriate. Our proposed system can be used instead in order to reduce their burden & allot marks equally. Online courses are getting popular among students. Trends of objective examination for such courses are already available but many courses require assessment in traditional way so that the subject understanding of the candidate can be evaluated which requires subjective assessment Le descriptive based examination

Presently, we have automated systems for objective type, single sentence answers and answers with less accuracy level. The system will evaluate the answer based upon the matched keywords and the minimum length of the answer provided by the moderator. The scanned handwritten answer sheet will be given as input to the proposed system. Along with the evaluation is a very tedious work and very time-consuming Thus in this project we are focusing on the inference process required for development of such type of systems. Our system is able to evaluate one word and one sentence-based answer with more than 80% of efficiency.

The idea of this project is to propose a system for online paper evaluation using machine learning and natural language processing for typed/handwritten answer sheets.
The input answer sheet by the students have to undergo a similarity check with the scheme of evaluation both syntactically and semantically.

ACKNOWLEDGEMENT

The magnitude of this project demanded the co-operation, guidance and assistance from a number of people and by the grace of God we have been fortunate enough to have this in entire process of completion of our project work.

We would like to thank our Principal Dr. Sr. Lalitha Thomas for providing us the facilities to carry out the project work.

We thank Ms. Irene Getzi, Head of the Department of MCA, Jyoti Nivas College Autonomous Bangalore for having permitted us to take this project.

We would like to place on record the valuable guidance and constant encouragement by our guide Dr. Senthil Vadivu M, Department of MCA, Jyoti Nivas College Autonomous, Bangalore for the suggestion to give a nice shape to this project.

We would like to thank all teaching and non-teaching staff members of MCA Department, Jyoti Nivas College Autonomous for their co-operation and constant encouragement which helped us in successfully completing our Mini project.

Finally, our heartfelt gratitude to all those who have helped us directly and indirectly, thus making the project successful.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Overview and objectives

The user authentication is done through a special web page called Login Page. The Login page Asks you to enter your credentials which is then validated by the application and after successful validation you are presented with the secured part of the application.

1.2 Objectives

The login page allows a user to gain access to an application by entering their username and password or by authenticating.

Single Sign-On
Multi-Factor Login

Usage scenarios related to the login page include:
A user navigates to an application and is presented with a login page as a way to gain access to the application.
There are two possible results:
Authentication is successful and the user is directed to the application   landing page.
Authentication fails and the user remains on the login page. If authentication fails, the screen should show an informational or error message about the failure.
A user is automatically logged out due to inactivity. In this event, they will be returned to the login page.

A user has forgotten their username and/or password. A link is available to begin the process to reset this information. Once the user clicks on one of these links, the contents of the login page is replaced with fields specific to recovering their username and/or password. There are a number of different ways the user could recover their password. This pattern does not dictate which methods an application should follow. Some options include:

The user could provide their e-mail and be sent a temporary password or a link to reset their password.
The user could answer a security question.
The user could get a message explaining that they have to contact a specific person.

2. ENVIRONMENT SPECIFICATION

2.1. Hardware Specifications

• MONITOR	:	LCD Monitor
• PROCESSOR	:	Intel® Core™ i5
• RAM	:	8.00 GB
• HARD DISK	:	64 GB or more

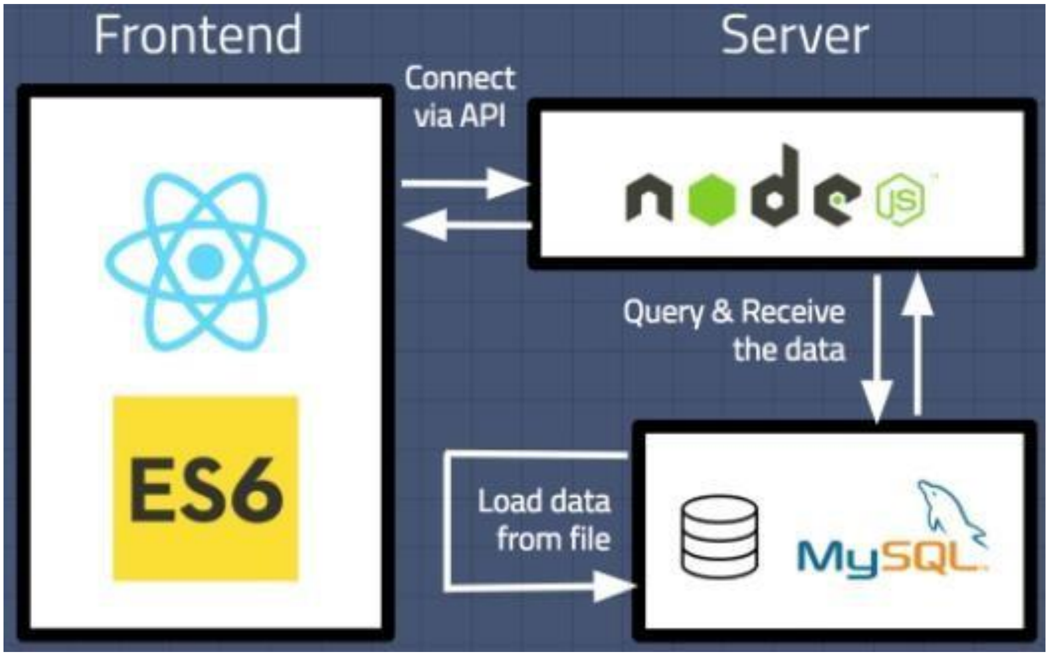
2.2. Software Specifications

• OPERATING SYSTEM	:	Windows 10
• PLATFORM	:	Visual Studio Code
• CODING LANGUAGE	:	Angular,Html,Css
• BACK END	:	MySql Workbench 8.0 CE
	:	

3. SOFTWARE DEVELOPMENT METHODOLOGY

3.1. Architectural Diagram

Login Module Process flow diagram



3.2. Functional requirements

User Stories with acceptance criteria

1. **As an Admin, I must be able to login with the credentials.**

   Scenario: Admin provides correct email and password.
   Given: Admin is in login page.
   When: Admin enter their email and password correctly.
   And: Click 'login'
   Then: Admin is taken to home page.

   Scenario: Admin does not provide correct email and password.
   Given: Admin is in login page.
   When: Admin enter their email and password incorrectly.
   And: Click 'login'
   Then: Admin see a message "Incorrect email or password".

   Given: Admin is not in login page.
   When: Admin tries to access control.
   Then: Admin is asked to login.

2. **As an admin I want to click on reset password option in case I forget password so that I can reset my password securely.**

   Scenario: Forgot password?
   Given: Admin navigates to the login page.
   When: Admin selects <forgot password> option
   And: Enter the valid email to receive a link for password recovery
   Then: The system sends the link to the entered email

   Given: Admin receives link via mail
   When: The admin navigates through the link received in email.
   Then: The system enables the admin to set the new password.

   Scenario: Forgot password? Invalid email.
   Given: Admin navigates to the login page.
   When: Admin selects <forgot password> option
   And: Enter the Invalid email.
   Then: pops up a message stating Invalid Email.

   Scenario: Confirm Password successful.
   Given: Admin is in create new password page.
   When: Admin enter same password for new password and confirm password.
   Then: pops up the message stating changed the password successfully.

   Scenario: error for Confirm Password
   Given: Admin is in create new password page.
   When: Admin enter different password for new password and confirm password.

Then: pops up the message stating “enter same password”.

3. **As a teacher, I would like to login with the credentials provided while registering.**

Scenario: Teacher provides correct email and password.
Given: Teacher is in login page.
When: Teacher enter their email and password correctly.
And: Click ‘login’
Then: Teacher is taken to home page.

Scenario: Teacher does not provide correct email and password.
Given: Teacher is in login page.
When: Teacher enter their email and password incorrectly.
And: Click ‘login’
Then: Teacher see a message “Incorrect email or password”.

Scenario: Teacher is not registered or verified.
Given: Teacher is in login page.
When: Teacher tries to login.
Then: Pop up a message stating that the teacher is not registered or verified.

Given: Teacher is not in login page.
When: Teacher tries to access control.
Then: Teacher is asked to login.

4. **As a teacher I want to click on reset password option in case I forget password so that I can reset my password securely.**

Scenario: Forgot password?
Given: Teacher navigates to the login page.
When: Teacher selects <forgot password> option
And: Enter the valid email to receive a link for password recovery
Then: The system sends the link to the entered email

Given: Teacher receives link via mail
When: The Teacher navigates through the link received in email.
Then: The system enables the Teacher to set the new password.

Scenario: Forgot password?
Given: Teacher navigates to the login page.
When: Teacher selects <forgot password> option
And: Enter the Invalid email.
Then: pops up a message stating Invalid Email.

Scenario: Confirm Password
Given: Teacher is in create new password page.
When: Teacher enter same password for new password and confirm password.

Then: pops up the message stating changed the password successfully.

Scenario: Confirm Password
Given: Teacher is in create new password page.
When: Teacher enter different password for new password and confirm password.
Then: pops up the message stating enter same password.

5. **As a student, I would like to login with the credentials provided while registering.**

Scenario: Student provides correct email and password.
Given: Student is in login page.
When: Student enter their email and password correctly.
And: Click 'login'
Then: Student is taken to home page.

Scenario: Student does not provide correct email and password.
Given: Student is in login page.
When: Student enter their email and password incorrectly.
And: Click 'login'
Then: Student see a message "Incorrect email or password".

Scenario: Student is not registered or verified.
Given: Student is in login page.
When: Student tries to login.
Then: Pop up a message stating that the teacher is not registered or verified.

Given: Student is not in login page.
When: Student tries to access control.
Then: Student is asked to login.

6. **As a student I want to click on reset password option in case I forget password so that I can reset my password securely.**

Scenario: Forgot password?
Given: Student navigates to the login page.
When: Student selects <forgot password> option
And: Enter the valid email to receive a link for password recovery
Then: The system sends the link to the entered email

Given: Student receives link via mail
When: The Student navigates through the link received in email.
Then: The system enables the student to set the new password.

Scenario: Forgot password?
Given: Student navigates to the login page.
When: Student selects <forgot password> option
And: Enter the Invalid email.
Then: pops up a message stating Invalid Email.

Scenario: Confirm Password
Given: Student is in create new password page.
When: Student enter same password for new password and confirm password.
Then: pops up the message stating changed the password successfully.

Scenario: Confirm Password
Given: Student is in create new password page.
When: Student enter different password for new password and confirm password.
Then: pops up the message stating enter same password.

3.3. Non-Functional Requirements

- **Usability:** The developed API is easy to work with as the user just has to enter the email and password to verify.
- **Security:** The API is developed in such a way that there is no threat of the password getting leaked as the password is hashed and then stored in the database.
- **Correctness:** The API respects the specifications i.e. only the registered and verified users are allowed to login.

4. SYSTEM DESIGN

System Design

Design is a place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps: Preliminary design is concerned with the transformation of requirements into data.

4.1 E-R Diagrams

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be located in one building.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models

One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database and each physical ER model is technology dependent since each database management system is somewhat different .

Conceptual data model:

This is the highest level ER model in that it contains the least granular detail but establishes the overall scope of what is to be included within the model set. The conceptual ER model normally defines master reference data entities that are commonly used by the organization. Developing an enterprise-wide conceptual ER model is useful to support documenting the data architecture for an organization.

A conceptual ER model may be used as the foundation for one or more logical data models (see below). The purpose of the conceptual ER model is then to establish structural metadata commonality for them aster data entities between the set of logical ER models. The conceptual data model may be used to form commonality relationships between ER models as a basis for data model integration.

Logical data model:

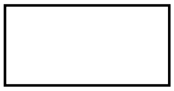
A logical ER model does not require a conceptual ER model, especially if the scope of the logical ER model includes only the development of a distinct information system.

Physical data model:

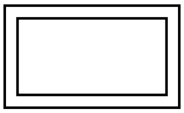
One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database.

Notations for ER diagram

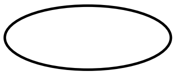
- Entity



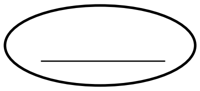
- Weak entity



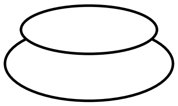
- Attribute



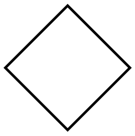
- Key attribute



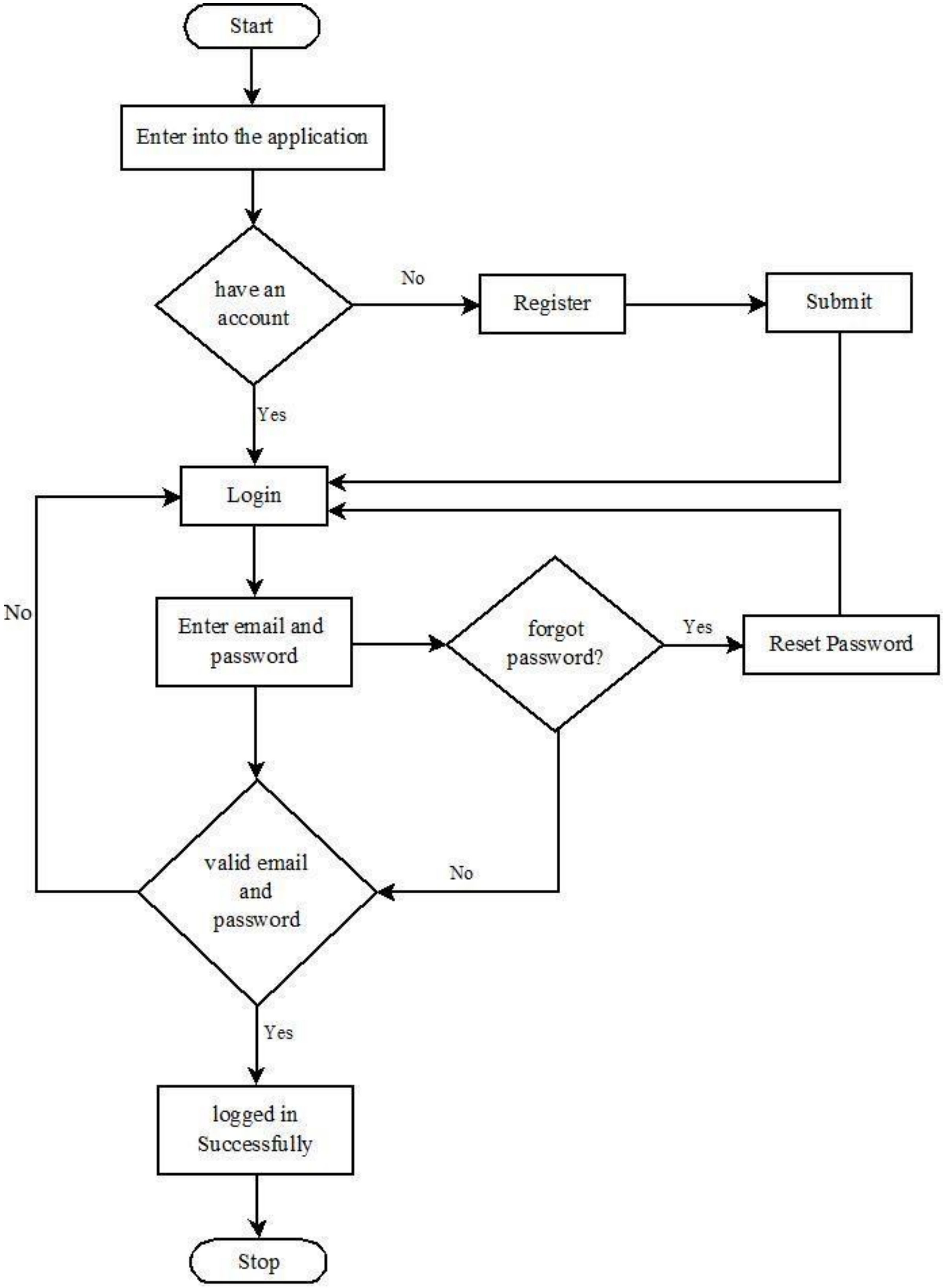
- Multivalued attribute



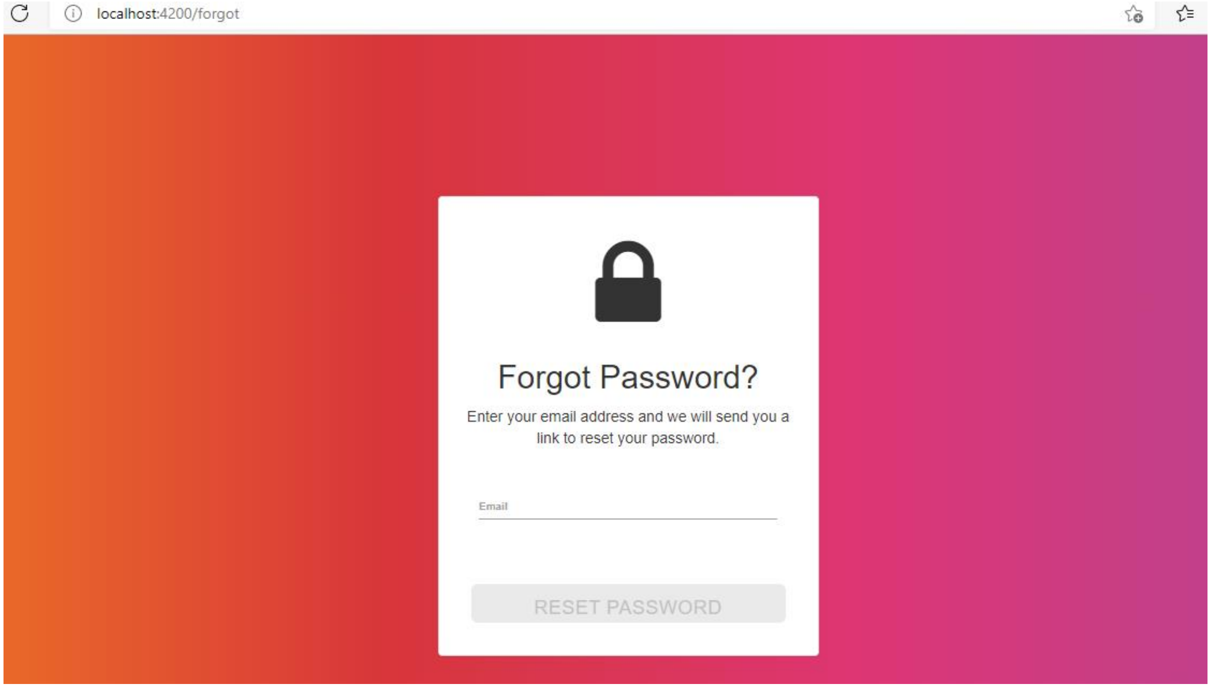
- Relation



Login page ER diagram



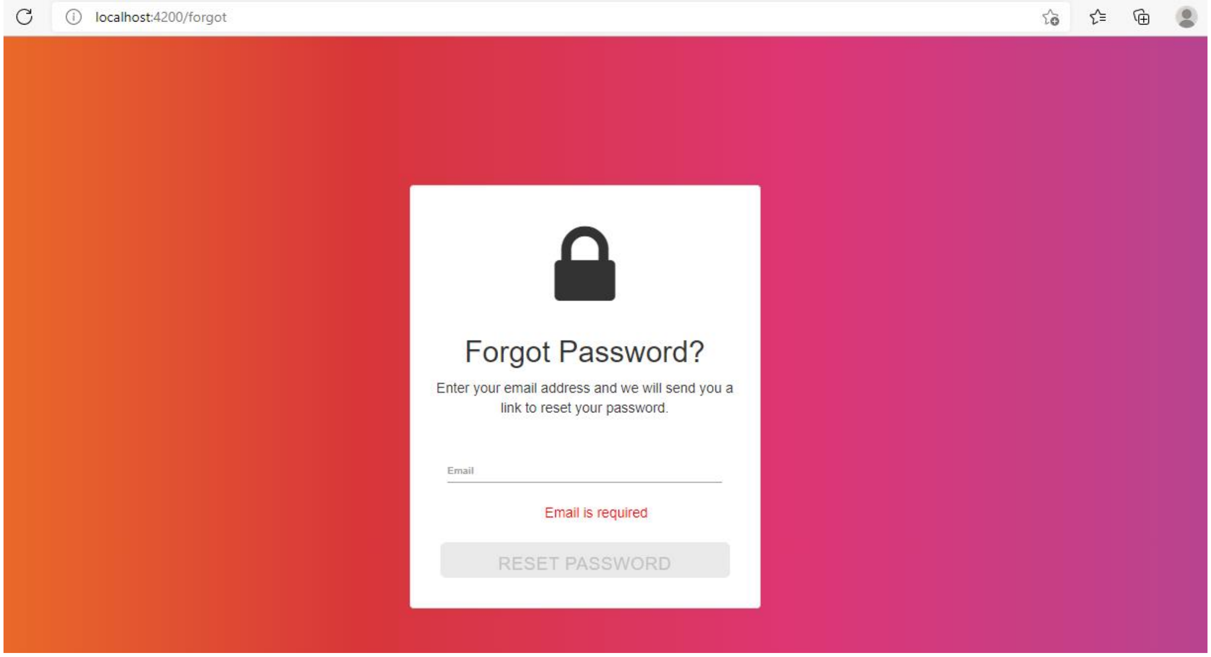
4.3 Interface design



localhost:4200/forgot

Forgot Password?

Enter your email address and we will send you a link to reset your password.

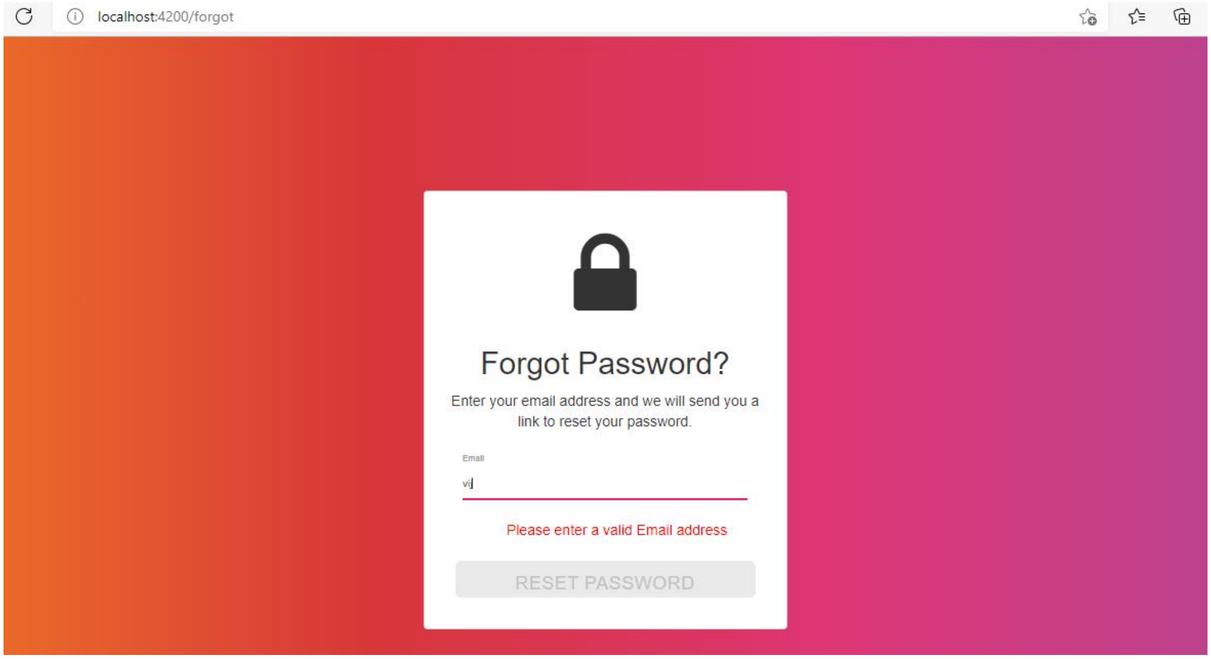


localhost:4200/forgot

Forgot Password?

Enter your email address and we will send you a link to reset your password.

Email is required

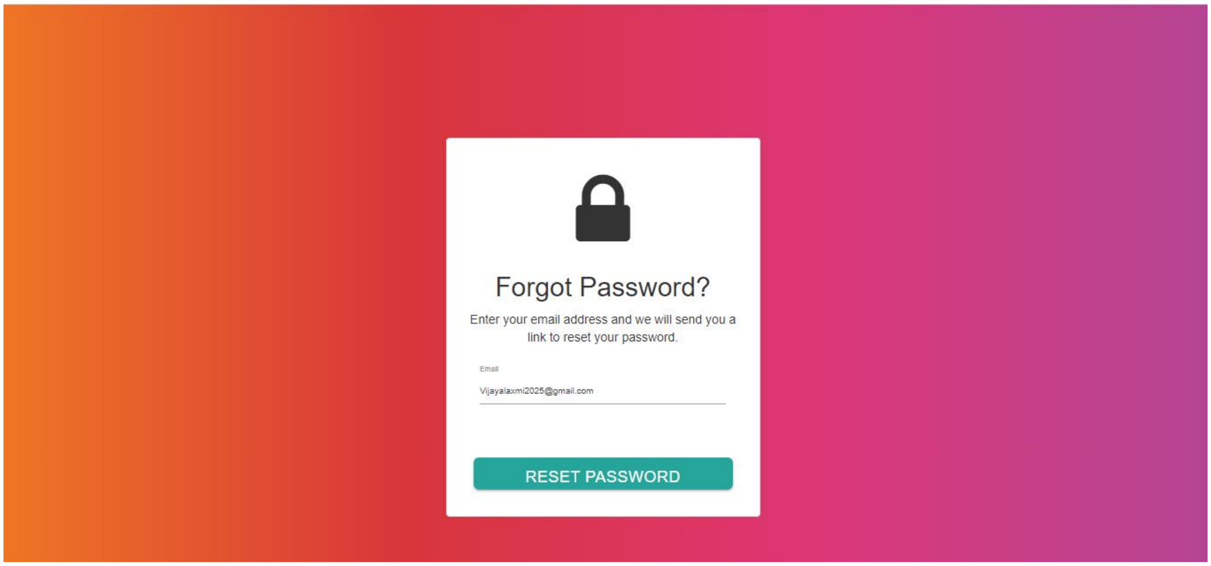


Forgot Password?

Enter your email address and we will send you a link to reset your password.

Email: vi

Please enter a valid Email address



Forgot Password?

Enter your email address and we will send you a link to reset your password.

Email: Vijayalaxmi2025@gmail.com

5. SYSTEM IMPLEMENTATION

System implementation:-

Implementation also involves packaging, handling, and storage, depending on the concerned technologies and where or when the system requirement needs to be integrated into a higher-level aggregate. Developing the supporting documentation for a system requirement, such as the manuals for operation, maintenance, and/or installation, is also a part of the implementation process; these art if acts are utilized in the system deployment and use phase. The system element requirements and the associated verification and validation criteria are inputs to this process; these inputs come from the architectural design process detailed outputs.

Execution of the implementation process is governed by both industrial and government standards and the terms of all applicable agreements. This may include conditions for packaging and storage, as well as preparation for use activities, such as operator training. In addition, packaging, handling, storage, and transportation (PHS&T) considerations will constrain the implementation activities. For more information, refer to the discussion of PHS&T in the System Deployment and Use article. The developing or integrating organization will likely have enterprise-level safety practices and guidelines that must also be considered.

Activities of the Process
The following major activities and tasks are performed during this process:
• Define the implementation strategy - Implementation process activities begin with detailed design and include developing an implementation strategy that defines fabrication and coding procedures, tools and equipment to be used, implementation tolerances, and the means and criteria for auditing configuration of resulting elements to the detailed design documentation. In the case of repeated system element implementations (such as for mass manufacturing or replacement elements), the implementation strategy is defined and refined to achieve consistent and repeatable element production; it is retained in the project decision database for future use. The implementation strategy contains the arrangements for packing, storing, and supplying the implemented element.

• Realize the system element - Realize or adapt and produce the concerned system element using the implementation strategy items as defined above. Realization or adaptation is conducted with regard to standards that govern applicable safety, security, privacy, and environmental guidelines or legislation and the practices of the relevant implementation technology. This requires the fabrication of hardware elements, development of software elements, definition of training capabilities, drafting of training documentation, and the training of initial operators and maintainers.
• Provide evidence of compliance - Record evidence that the system element meets its requirements and the associated verification and validation criteria as well as the legislation

5.1 Tools and Technologies

5.1.1 Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

5.1.2 Angular JS:-

AngularJS is an open-source web application framework. AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application components clearly and succinctly. Its data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.
The general features of AngularJS are as follows −
AngularJS is a efficient framework that can create Rich Internet Applications (RIA).
AngularJS provides developers an options to write client side applications using JavaScript in a clean Model View Controller (MVC) way.
Applications written in AngularJS are cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache license version 2.0.
Overall, AngularJS is a framework to build large scale, high-performance, and easyto-maintain web applications.

The core features of AngularJS are as follows :-

Data-binding − It is the automatic synchronization of data between model and view components.

Scope − these are objects that refer to the model. They act as a glue between controller and view.

Controller −These are JavaScript functions bound to a particular scope.

Services − AngularJS comes with several built-in services such as $http to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.

Filters − these select a subset of items from an array and returns a new array.

Directives − Directives are markers on DOM elements such as elements, attributes, css, and more. These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives such as ngBind, ngModel, etc.

Templates − These are the rendered view with information from the controller and model. These can be a single file (such as index.html) or multiple views in one page using partials.

Routing − It is concept of switching views.

Model View Whatever − MVW is a design pattern for dividing an application into different parts called Model, View, and Controller, each with distinct responsibilities. AngularJS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-View-View Model). The Angular JS team refers it humorously as Model View Whatever.

Deep Linking − Deep linking allows to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.
Dependency Injection − AngularJS has a built-in dependency injection subsystem that helps the developer to create, understand, and test the applications easily.

5.1.3 Node.js

Node.js (Node) is an open source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications.
Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.
Following are the areas where Node.js is proving itself as a perfect technology partner.
I/O bound Applications
Data Streaming Applications
Data Intensive Real-time Applications (DIRT)
JSON APIs based Applications
Single Page Applications

5.1.4.HTML 5

HTML 5is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard.
It was published in October 2014 by the World Wide Web Consortium to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.
HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind. Many new syntactic features are included.

The APIs and Document Object Model (DOM) are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for any invalid documents.
The Web Hypertext Application Technology Working Group (WHATWG) began work on the new standard in 2004. At that time, HTML 4.01 had not been updated since 2000, and the World Wide Web Consortium (W3C) was focusing future developments onXHTML 2.0. In 2009, the W3C allowed the XHTML 2.0 Working Group's charter to expire and decided not to renew it.[8] W3C and WHATWG are currently working together on the development of HTML5.

While some features of HTML5 are often compared to Adobe Flash, the two technologies are very different. Both include features for playing audio and video within web pages, and for using Scalable Vector Graphics. However, HTML5 on its own cannot be used for animation or interactivity – it must be supplemented with CSS3 or JavaScript. There are many Flash capabilities that have no direct counterpart in HTML5. See Comparison of HTML5 and Flash.

5.1.5 CSS 3 :-

Unlike CSS 2, which is a large single specification defining various features, CSS 3 is divided into several separate documents called "modules". Each module adds new capabilities or extends features defined in CSS 2, preserving backward compatibility. Work on CSS level 3 started around the time of publication of the original CSS 2 recommendation. The earliest CSS 3 drafts were published in June 1999.

Due to the modularization, different modules have different stability and statuses. As of June 2012, there are over fifty CSS modules published from the CSS Working Group. and four of these have been published as formal recommendations:

Further information: Comparison of layout engines (Cascading Style Sheets)
Because not all browsers correctly parse CSS code, developed coding techniques known as CSS hacks can either filter specific browsers or target specific browsers (generally both are known as CSS filters). The former can be defined as CSS filtering hacks and the latter can be defined as CSS targeting hacks. Both can be used to hide or show parts of the CSS to different browsers. This is achieved either by exploiting CSS-handling quirks or bugs in the browser, or by taking advantage of lack of support for parts of the CSS specifications. Using CSS filters, some designers have gone as far as delivering different CSS to certain browsers to ensure designs render as expected. Because very early web browsers were either completely incapable of handling CSS, or rendered CSS very poorly, designers today often routinely use CSS filters that completely prevent these browsers from accessing any of the CSS. Internet Explorer was the first browser to provide support for CSS, with IE 3.0, and increased progressively with each version. By 2008, the first Beta of Internet Explorer 8 offered support for CSS 2.1 in its best web standards mode.

5.1.6 JavaScript

JavaScript is a high-level, dynamic, untyped, and interpreted programming language. It Alongside HTML and CSS, it is one of the three core technologies of World Wide Webcontent production; the majority of websites employ it and it is supported by all modern Web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, andfunctional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two are distinct languages and differ greatly in their design. JavaScript was influenced by programming languages such as Self and Scheme. JavaScript is also used in environments that are not Web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side Web applications. On the client side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation. It is also used in game development, the creation of desktop and mobile applications, and server-side network programming with run-time environments such as Node.js

5.1.7 Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to twitter developer Mark Otto: "A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company."

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011 It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

On January 31, 2012, Bootstrap 2 was announced. This release added the twelve-column grid layout and responsive design components, as well as changes to many of the existing components.

The Bootstrap 3 release was announced on 19 August 2013, moving to a mobile first approach and using a flat design

3. CONCLUSIONS AND FUTURE SCOPE

CONCLUSION

With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Mini Project ”Automatic Descriptive Answer Evaluation System (Login Page)” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

The planning that goes into implementing a project.
The importance of proper planning and an organized methodology.
The key element of team spirit and coordination in a successful project.

The project also provided us the opportunity of interacting with our teachers and to gain from their best experience

FUTURE SCOPE

This software has been developed in such a way that it can accept modifications and further changes. The software is very user friendly and further changes can be done easily.
Software restricting is carried out. Software restricting modifies source code in an effort to make it meaningful for future changes. In general restructuring does not modify the overall program architecture.
Every system should allow scope for further development or enhancement. The system can be adapted for any further development. The system is so flexible that it allows for any modification needed for further functioning of programs.
Since the objectives may be brought broad in future, the system can be easily modified accordingly, as the system has been modularized. The future expansion can be done in a concise manner in order to improve the efficiency of the system.

APPENDIX

Source Code

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Login</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=s
wap" rel="stylesheet">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/css/materialize
.min.css">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
</head>
<body class="gradient-custom-2">
  <app-root></app-root>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/js/materialize.m
in.js"></script>

</body>
</html>
```

app.component.html

```
<router-outlet></router-outlet>
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { RouterModule } from '@angular/router';
import { appRoutes } from './routes';
import { FormsModule, ReactiveFormsModule} from '@angular/forms';
import { ForgotPasswordComponent } from './forgot-password/forgot-
password.component';

import { ChangePasswordComponent } from './change-password/change-
password.component';
import { StudentSinginComponent } from './student-singin/student-
singin.component';
import { MatCardModule } from '@angular/material/card';
import {MatFormFieldModule} from '@angular/material/form-field';

@NgModule({
  declarations: [
    AppComponent,
    ForgotPasswordComponent,
    ChangePasswordComponent,
    StudentSinginComponent,

    ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    FormsModule,
    RouterModule.forRoot(appRoutes),
    ReactiveFormsModule,
    MatCardModule,
    MatFormFieldModule
  ],
  providers: [],
  bootstrap: [AppComponent]

})
export class AppModule { }
```

Routes.ts

```
import { Routes } from '@angular/router'

import { ForgotPasswordComponent } from './forgot-password/forgot-
password.component';
import { ChangePasswordComponent } from './change-password/change-
password.component';

import { StudentSinginComponent } from './student-singin/student-
singin.component';

export const appRoutes: Routes = [

    {   path: 'forgot', component: ForgotPasswordComponent },
    {   path: 'change', component: ChangePasswordComponent },
     {   path: 'stu', component: StudentSinginComponent },

    { path : '', redirectTo:'/stu', pathMatch : 'full'}

];
```

forgot-password.component.css:-

```
.form-gap {
    padding-top: 150px;
}
.mat-hint{
    color: red;
    text-size-adjust: 10px;
    text-align: center;
    }
.lab{
    text-size-adjust: 80%;
}
```

forgot-password.component.html:-

```
<link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css" rel="stylesheet"
id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-
awesome.min.css">

<div class="form-gap"></div>
<div class="container">
            <div class="row">
                        <div class="col-md-4 col-md-offset-4">
            <div class="panel panel-default">
              <div class="panel-body">
                <div class="text-center">
                  <h3><i class="fa fa-lock fa-4x"></i></h3>
                  <h2 class="text-center">Forgot Password?</h2>
                  <p>Enter your email address and we will send you a link to reset your
password.</p>
                  <div class="panel-body">

                    <form [formGroup]="Forgotform"  id="Forgot-form" role="form"
autocomplete="off" class="form" method="post" >

                        <div class="input-field col s12">
                                  <input type="email" #Email ngModel name="Email"
formControlName="email"/>
                                  &nbsp; &nbsp; &nbsp;
```

```
                              <mat-hint *ngIf="Forgotform.touched &&
Forgotform.get('email').hasError('required')"> Email is required  </mat-hint>
                              <mat-hint *ngIf="Forgotform.touched &&
Forgotform.get('email').hasError('email')">Please enter a valid Email address</mat-hint>
                              <label class="active" ><b>Email</b></label>
                     </div>

                   &nbsp; &nbsp; &nbsp;



<button [disabled]="!Forgotform.valid" name="recover-submit" class="btn btn-lg btn-block "
type="submit">Reset Password</button>

         </form>
           </div>
              </div>
            </div>
           </div>
         </div>
           </div>
           </div>
```

forgot-password.component.ts:-

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { ForgotPasswordComponent } from './forgot-password.component';

describe('ForgotPasswordComponent', () => {
  let component: ForgotPasswordComponent;
  let fixture: ComponentFixture<ForgotPasswordComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ForgotPasswordComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(ForgotPasswordComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

forgot-password.component.html:-

```
import { Component, OnInit } from '@angular/core';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';
@Component({
  selector: 'app-forgot-password',
  templateUrl: './forgot-password.component.html',
  styleUrls: ['./forgot-password.component.css']
})
export class ForgotPasswordComponent implements OnInit {

  Forgotform: FormGroup;
  constructor(private fb: FormBuilder) {


    this.setupForm();
  }
  setupForm(){
    this.Forgotform=this.fb.group({
      email:['',[Validators.required, Validators.email]],
          });

  }
  ngOnInit(): void {
  }

}
```

styles.css

```
/* You can add global styles to this file, and also import other style files */

html, body { height: 100%; }
body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif; }

/* You can add global styles to this file, and also import other style files */
button.btn-submit{
    background-color: #38547b;
    color: #fff;
    width: 100%;
}
button.btn-submit:focus,button.btn-submit:hover{
    background-color: #38547b;
}

input.ng-invalid.ng-dirty{
    border-bottom-color : #e91e63 !important;
    box-shadow: 0 1px 0 0 #e91e63 !important;
}
/*for error div i*/
```

```
div.error-message i{
    vertical-align: middle !important;
}

/*for tab control*/
.tabs{
    height: 65px;
}
.tabs .tab{
    text-transform : none !important;
    line-height: 65px;
    height: 65px;
}
.tabs .tab a{
    font-size: 25px;
    color: grey !important;
}

.tabs .tab a.active{
    background-color: #fff !important;
}
.text-center{
    text-align: center;
}
.gradient-custom-2 {
    /* fallback for old browsers */
    background: #fccb90;

    /* Chrome 10-25, Safari 5.1-6 */
    background: -webkit-linear-gradient(to right, #ee7724, #d8363a, #dd3675,
#b44593);

    /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */
    background: linear-gradient(to right, #ee7724, #d8363a, #dd3675, #b44593);
  }
```

REFERENCES

- https://www.tutorialspoint.com/angularjs/angularjs_login_application.htm
- https://krazytech.com/programs/simple-login-example-in-angularjs
- https://stackoverflow.com/questions/20969835/angularjs-login-and-authentication
- https://www.youtube.com/watch?v=mZnzX3J5XKI
- https://www.youtube.com/watch?v=e8BlURn6SFk&list=RDCMUCvzlnZbePin9kH-1JCKBt8Q&start_radio=1&rv=e8BlURn6SFk&t=34