**Week 9 & 10:**

**Create an application for Registration form.**

**Creating Form**

Flutter provides a **Form widget** to create a form. The form widget acts as a container, which allows us to group and validate the multiple form fields. When you create a form, it is necessary to provide the **GlobalKey**. This key uniquely identifies the form and allows you to do any validation in the form fields.

The form widget uses child widget **TextFormField** to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur.

Let us create a form. First, create a Flutter project. Inside the class, we define a global key as **\_formKey**. This key holds a **FormState** and can use to retrieve the form widget. Inside the **build** method of this class, we have added some custom style and use the TextFormField widget to provide the form fields such as name, Email Id, Password fields. Inside the TextFormField, we have used **InputDecoration** that provides the look and feel of your form properties such as borders, labels, icons, hint, styles, etc. Finally, we have added a **button** to submit the form.

**main.dart**

import 'package:flutter/material.dart';

import 'login\_screen.dart';

void main()

{

var app=MaterialApp(

home:LoginScreen(),

title:"flutter form"

);

runApp(app);

}

**registration.dart**

import 'package:flutter/material.dart';

import 'login.dart';

class LoginScreen extends StatefulWidget

{

const LoginScreen({super.key});

@override

State<LoginScreen> createState() => \_LoginScreenState();

}

class \_LoginScreenState extends State<LoginScreen>

{

final \_formKey = GlobalKey<FormState>(); // Step 1

bool isObsecureText = true;

TextEditingController \_nameController = TextEditingController();

TextEditingController \_emailControlle = TextEditingController();

TextEditingController \_passwordController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("User Registration"),

backgroundColor: Colors.lightBlue,

),

body: Container(

margin: EdgeInsets.only(left: 20, right: 20, top: 20),

width: MediaQuery.of(context).size.width,

height: MediaQuery.of(context).size.height,

child: Form(

key: \_formKey,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Image.asset("images/login.png"),

Text("Register Here!",

style: TextStyle(fontSize: 32, fontWeight: FontWeight.w600)),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(" Full Name"),

SizedBox(

height: 8,

),

TextFormField(

controller: \_nameController,

keyboardType: TextInputType.name,

validator: (value) {

print("Name text $value");

final inputValue = value ?? "";

if (inputValue.isEmpty) {

return "Please Enter Name";

} else {

return null;

}

},

decoration: InputDecoration(

hintText: "Please Enter Full Name",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

borderSide: BorderSide(color: Colors.grey))),

),

],

),

SizedBox(

height: 10,

),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(" Email ID"),

SizedBox(

height: 8,

),

TextFormField(

controller: \_emailControlle,

keyboardType: TextInputType.emailAddress,

validator: (value) {

print("email text $value");

final inputValue = value ?? "";

if (inputValue.isEmpty) {

return "Please Enter Email";

} else if (!checkEmailValidation(inputValue)) {

return "Please Enter Valid Email";

} else {

return null;

}

},

decoration: InputDecoration(

hintText: "Please Enter Email",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

borderSide: BorderSide(color: Colors.grey))),

),

],

),

SizedBox(

height: 10,

),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text("Password"),

SizedBox(

height: 8,

),

TextFormField(

controller: \_passwordController,

validator: (value) {

print("password text $value");

final inputValue = value ?? "";

if (inputValue.isEmpty) {

return "Please Enter Password";

} else if (!validatePassword(inputValue)) {

return "Please Enter valid Password";

} else {

return null;

}

},

obscureText: isObsecureText,

obscuringCharacter: "\*",

maxLength: 10,

decoration: InputDecoration(

suffixIcon: isObsecureText

? GestureDetector(

onTap: (){

setState(() {

isObsecureText = false;

});

print(isObsecureText);

},

child: Icon(Icons.visibility\_off))

: GestureDetector(

onTap: (){

setState(() {

isObsecureText = true;

});

print(isObsecureText);

},

child: Icon(Icons.visibility)),

hintText: "Please Enter Password",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

borderSide: BorderSide(color: Colors.grey))),

),

],

),

Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

Expanded(

child: ElevatedButton(

onPressed: () {

\_nameController.clear();

\_emailControlle.clear();

\_passwordController.clear();

},

style: ButtonStyle(

backgroundColor: WidgetStateProperty.all<Color>(Colors.blue),

foregroundColor: WidgetStateProperty.all<Color>(Colors.white)

),

child: Text('Reset'),

),

),

Expanded(

child: ElevatedButton(

onPressed: () {

final result = \_formKey.currentState!.validate();

if(result) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) =>

NextPage()));

}

} ,

style: ButtonStyle(

backgroundColor: WidgetStateProperty.all<Color>(Colors.blue), // Change button color

foregroundColor: WidgetStateProperty.all<Color>(Colors.white)

),

child: Text('Register'),

),

),

// Add more buttons as needed

],

),

],

),

),

),

);

}

bool checkEmailValidation(String email)

{

return RegExp(

r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'\*+-/=?^\_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+")

.hasMatch(email);

}

bool validatePassword(String password)

{

// Regular expression for validating the password

final RegExp passwordRegExp = RegExp(

r'^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$');

// Check if the password matches the regular expression

return passwordRegExp.hasMatch(password);

}

}

class NextPage extends StatelessWidget

{

const NextPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Registration Page'),

backgroundColor: Colors.lightBlue,

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text('Thank you for your Registration with us!!',

style: TextStyle(

fontSize: 18,

fontWeight: FontWeight.bold,

color: Colors.red,

),

),

ElevatedButton(

onPressed: ()

{

Navigator.push(

context,

MaterialPageRoute(

builder: (context) =>

Login()));

},

style: ButtonStyle(

backgroundColor: WidgetStateProperty.all<Color>(Colors.blue),

foregroundColor: WidgetStateProperty.all<Color>(Colors.white)

),

child: const Text('Login Here!'),

),

],

),

),

);

}

}

**login.dart**

import 'package:flutter/material.dart';

class Login extends StatefulWidget

{

const Login({super.key});

@override

State<Login> createState() => \_LoginState();

}

class \_LoginState extends State<Login>

{

final \_formKey = GlobalKey<FormState>(); // Step 1

bool isObsecureText = true;

TextEditingController \_emailControlle = TextEditingController();

TextEditingController \_passwordController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("User Login"),

backgroundColor: Colors.lightBlue,

),

body: Container(

margin: EdgeInsets.only(left: 20, right: 20, top: 20),

width: MediaQuery.of(context).size.width,

height: MediaQuery.of(context).size.height,

child: Form(

key: \_formKey,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Image.asset("images/login.png"),

Text("Login Here!",

style: TextStyle(fontSize: 32, fontWeight: FontWeight.w600)),

SizedBox(

height: 30,

),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(" Email ID"),

SizedBox(

height: 8,

),

TextFormField(

controller: \_emailControlle,

keyboardType: TextInputType.emailAddress,

validator: (value) {

print("email text $value");

final inputValue = value ?? "";

if (inputValue.isEmpty) {

return "Please Enter Email";

} else if (!checkEmailValidation(inputValue)) {

return "Please Enter Valid Email";

} else {

return null;

}

},

decoration: InputDecoration(

hintText: "Please Enter Email",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

borderSide: BorderSide(color: Colors.grey))),

),

],

),

SizedBox(

height: 30,

),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text("Password"),

SizedBox(

height: 8,

),

TextFormField(

controller: \_passwordController,

validator: (value) {

print("password text $value");

final inputValue = value ?? "";

if (inputValue.isEmpty) {

return "Please Enter Password";

} else if (!validatePassword(inputValue)) {

return "Please Enter valid Password";

} else {

return null;

}

},

obscureText: isObsecureText,

obscuringCharacter: "\*",

maxLength: 10,

decoration: InputDecoration(

suffixIcon: isObsecureText

? GestureDetector(

onTap: (){

setState(() {

isObsecureText = false;

});

print(isObsecureText);

},

child: Icon(Icons.visibility\_off))

: GestureDetector(

onTap: (){

setState(() {

isObsecureText = true;

});

print(isObsecureText);

},

child: Icon(Icons.visibility)),

hintText: "Please Enter Password",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

borderSide: BorderSide(color: Colors.grey))),

),

],

),

SizedBox(

width: MediaQuery.of(context).size.width \* 0.8,

child: TextButton(

style: TextButton.styleFrom(

backgroundColor: Color.fromRGBO(69, 182, 74, 1),

shape: RoundedRectangleBorder(

borderRadius:

BorderRadius.circular(8), // Rounded corners

),

),

onPressed: () {

final result = \_formKey.currentState!.validate();

if(result) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) =>

NextPage()));

}

},

child: Text(

"Login",

style: TextStyle(

fontSize: 18,

fontWeight: FontWeight.w600,

color: Colors.white),

)),

)

],

),

),

),

);

}

bool checkEmailValidation(String email)

{

return RegExp(

r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'\*+-/=?^\_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+")

.hasMatch(email);

}

bool validatePassword(String password)

{

// Regular expression for validating the password

final RegExp passwordRegExp = RegExp(

r'^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$');

// Check if the password matches the regular expression

return passwordRegExp.hasMatch(password);

}

}

class NextPage extends StatelessWidget

{

const NextPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Login Page'),

backgroundColor: Colors.lightBlue,

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text('Thank You !!',

style: TextStyle(

fontSize: 18,

fontWeight: FontWeight.bold,

color: Colors.red,

),

),

],

),

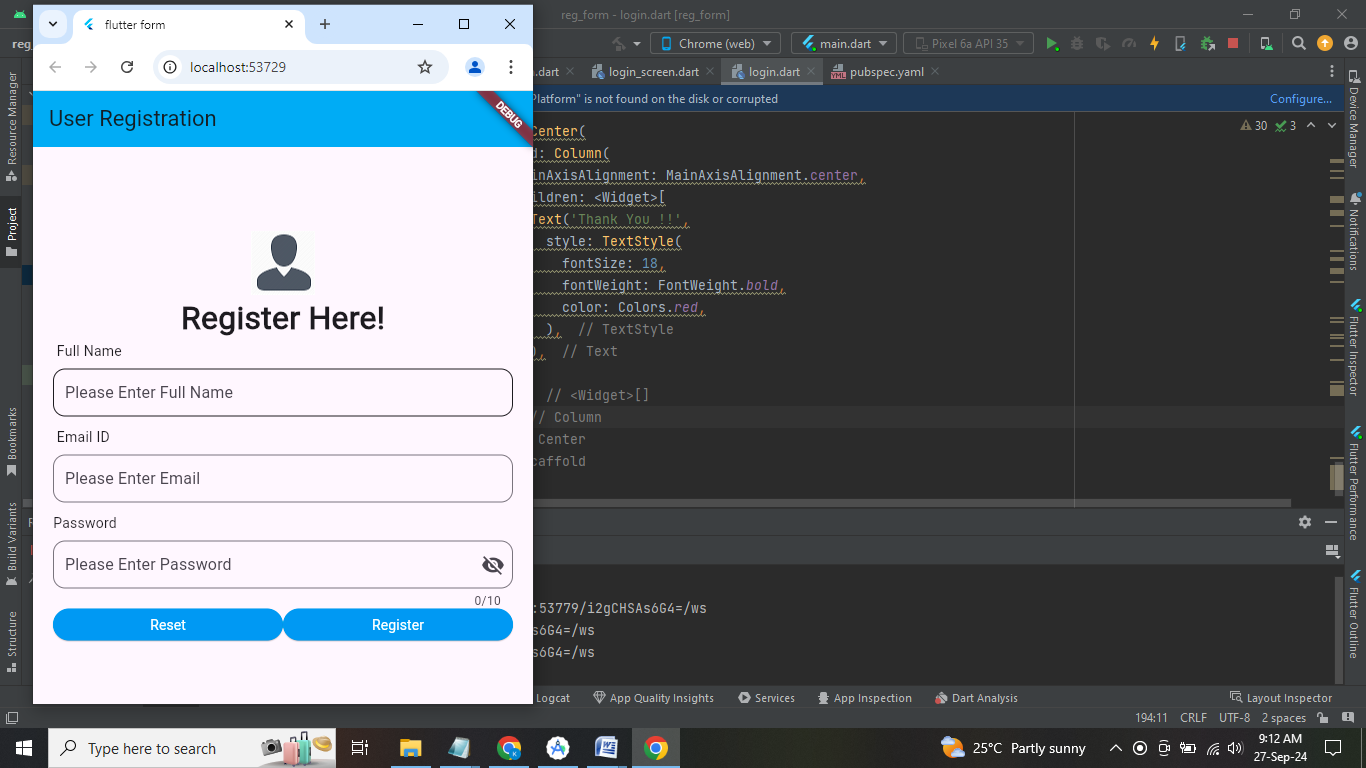
),

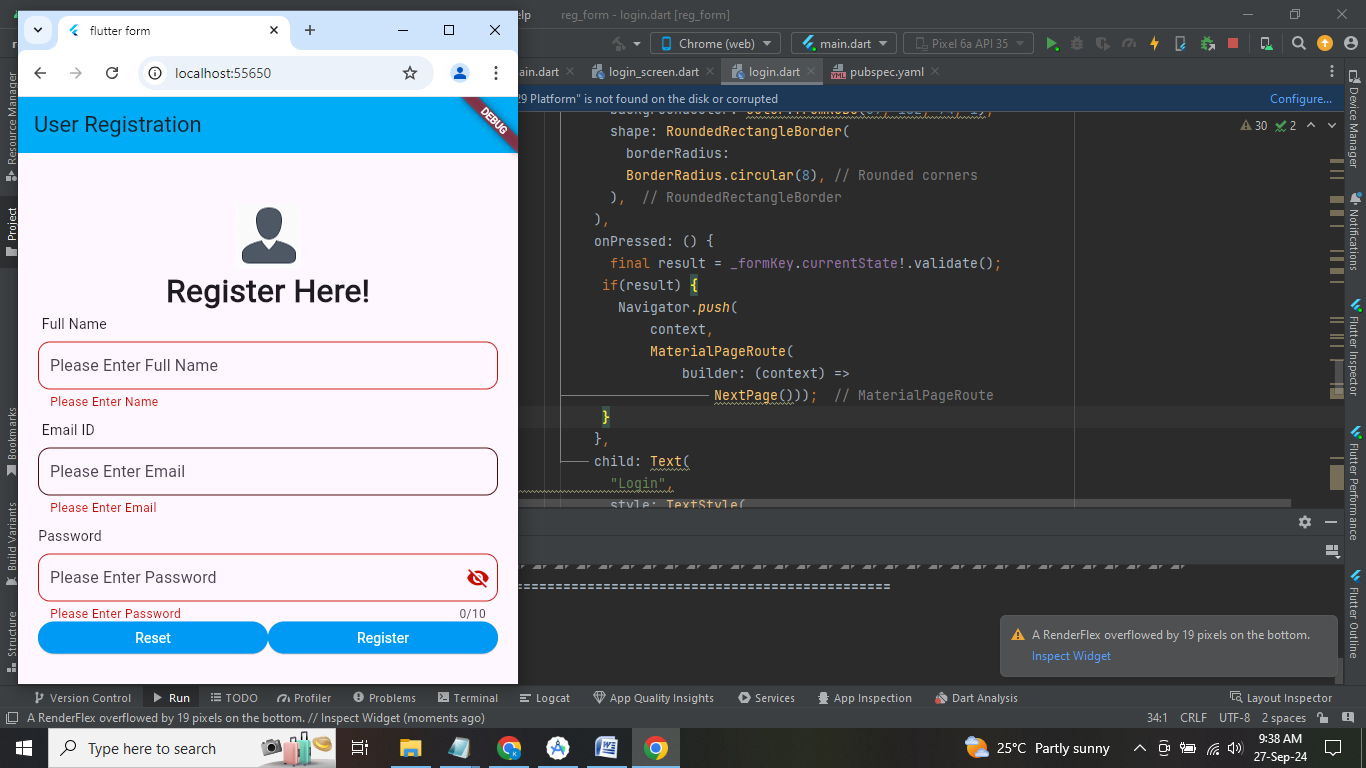
);

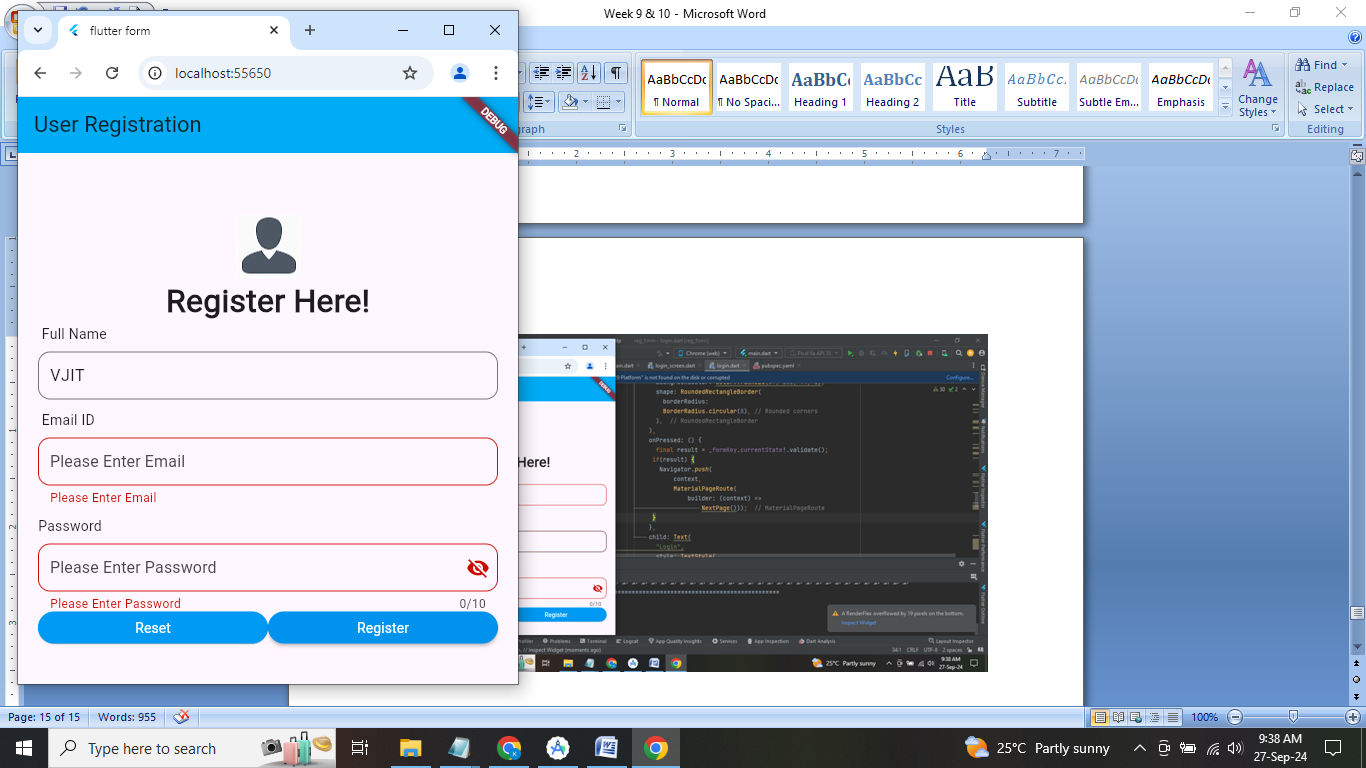
}

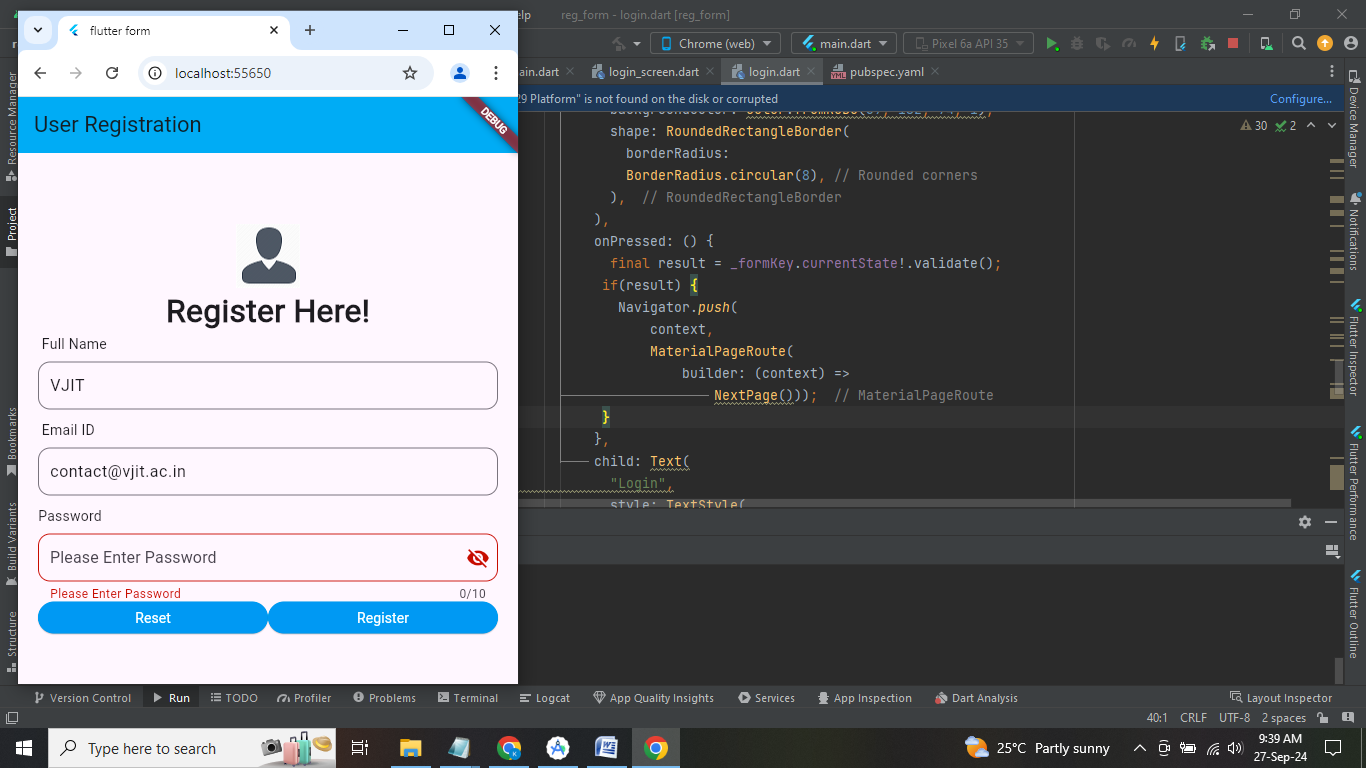
}

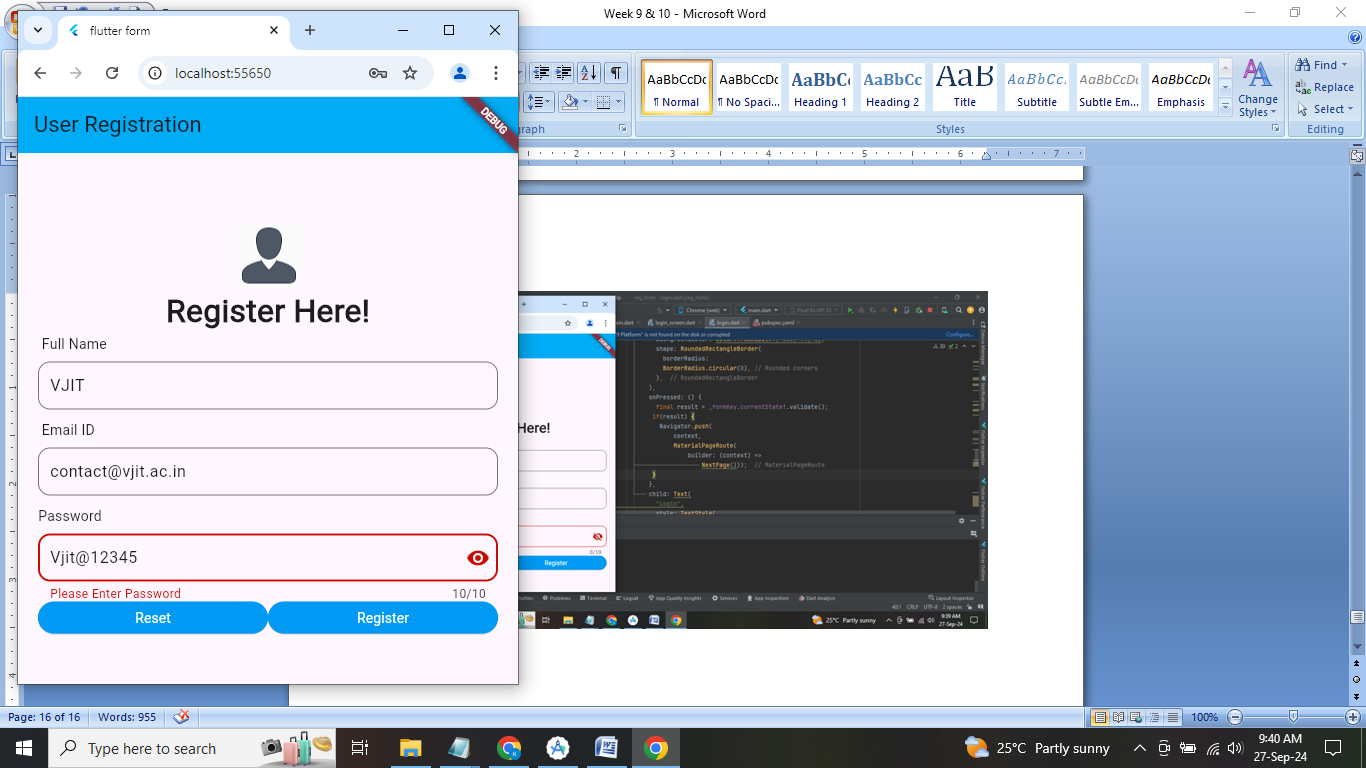
**OUTPUT:**

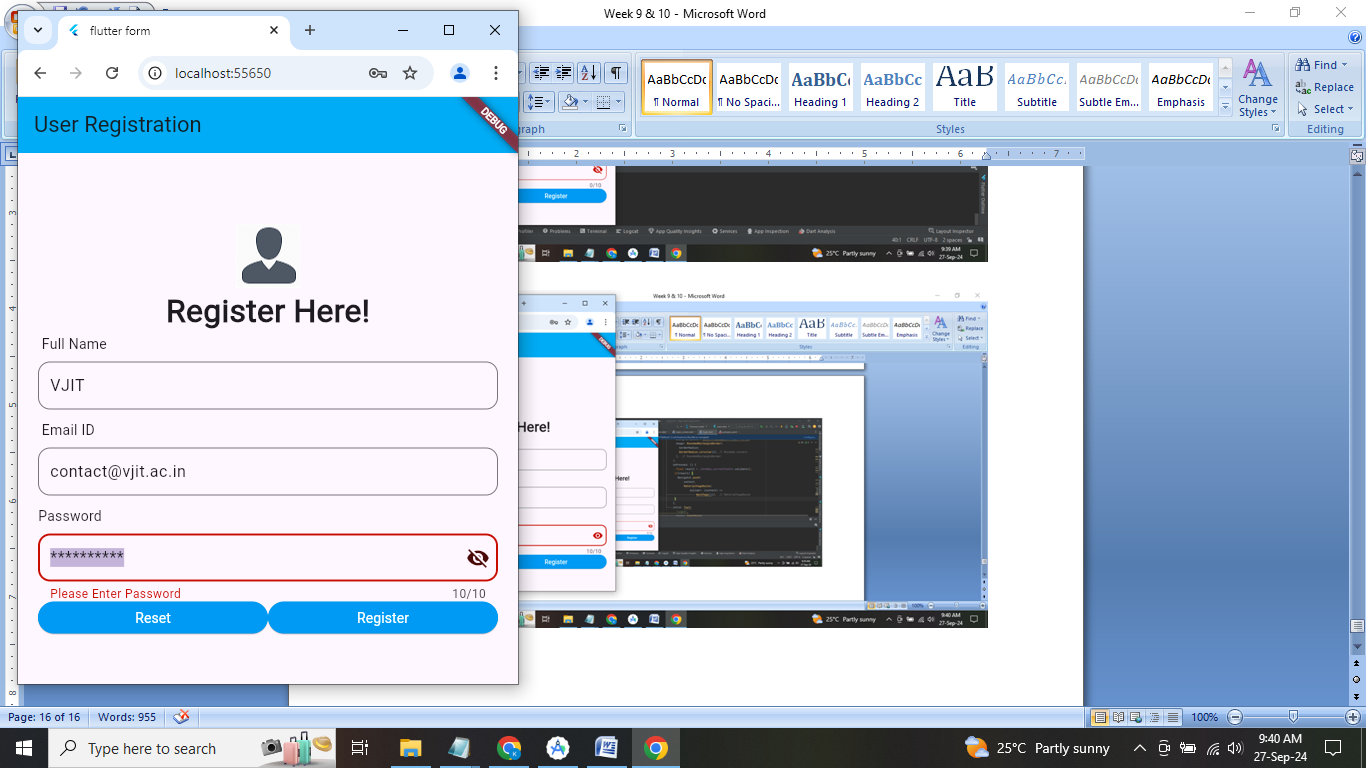
****

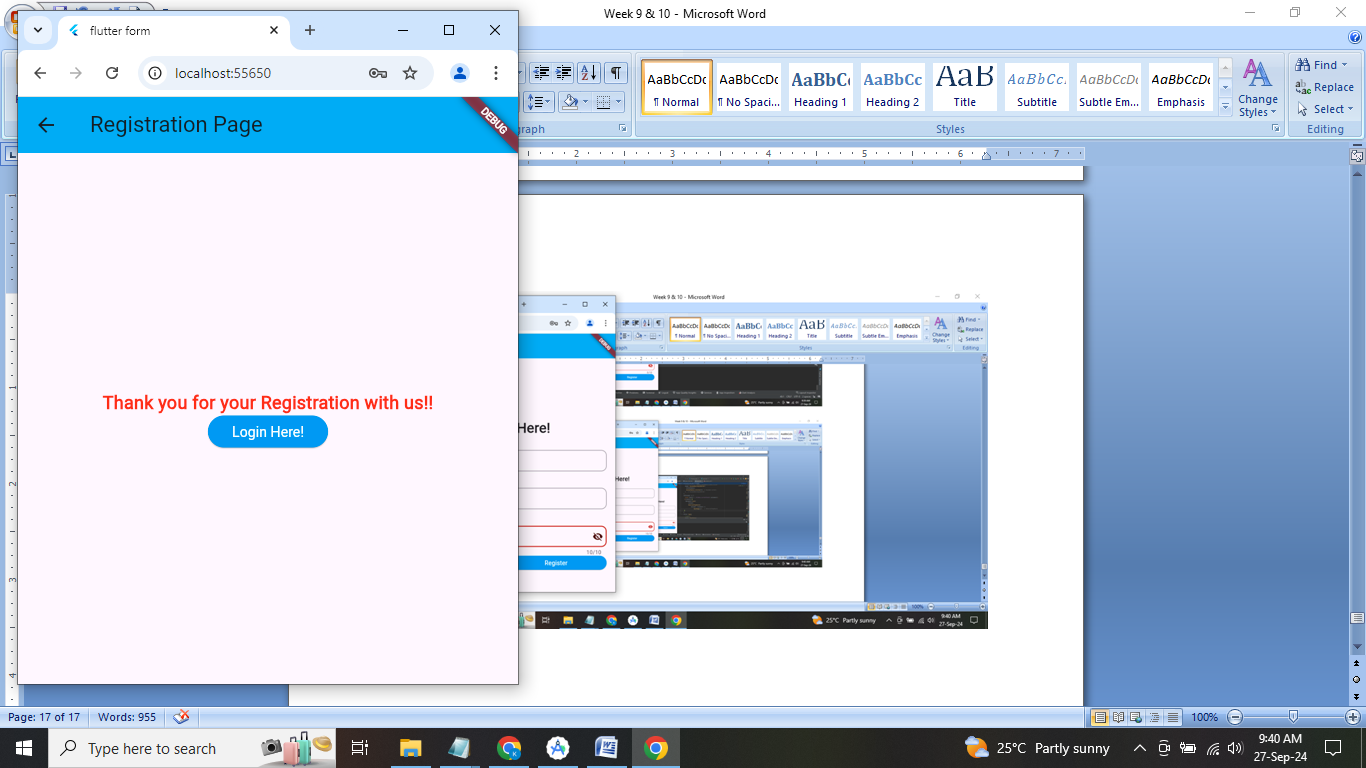
****

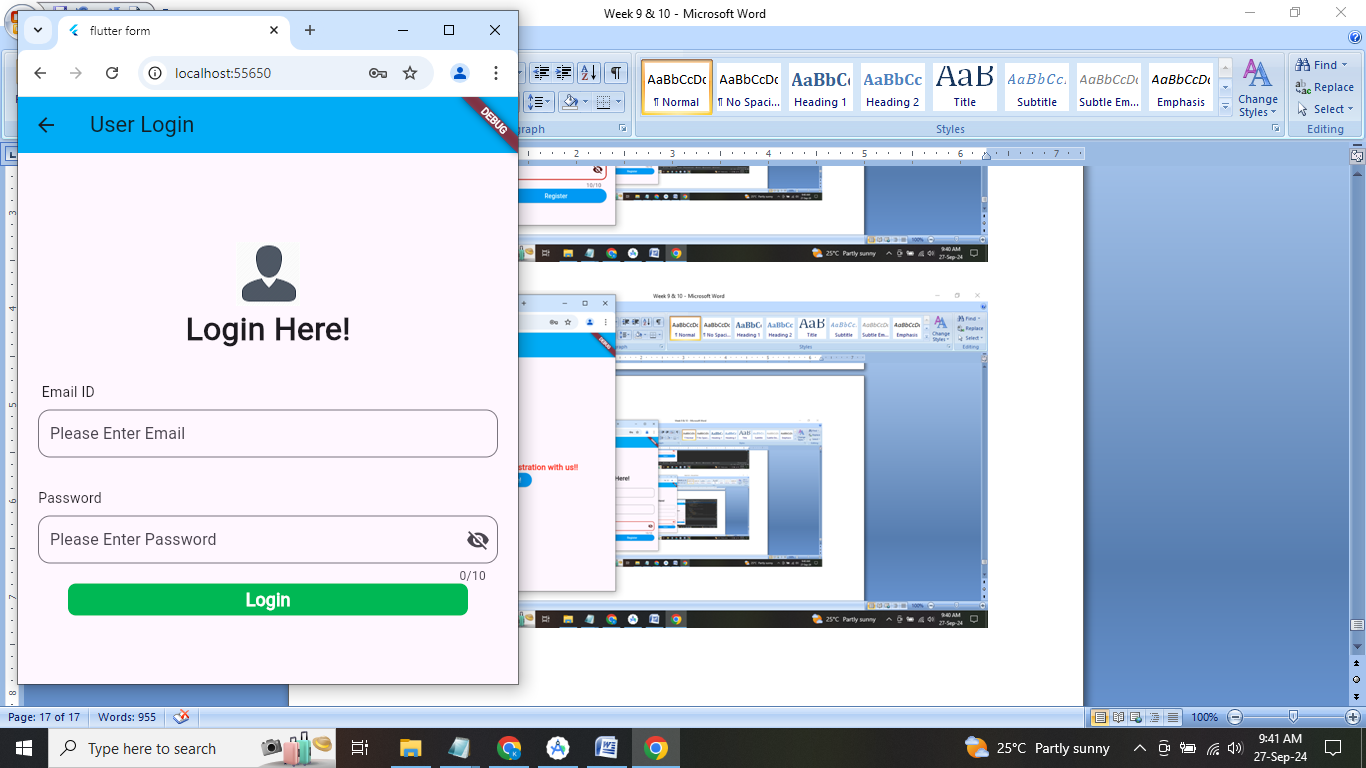
****

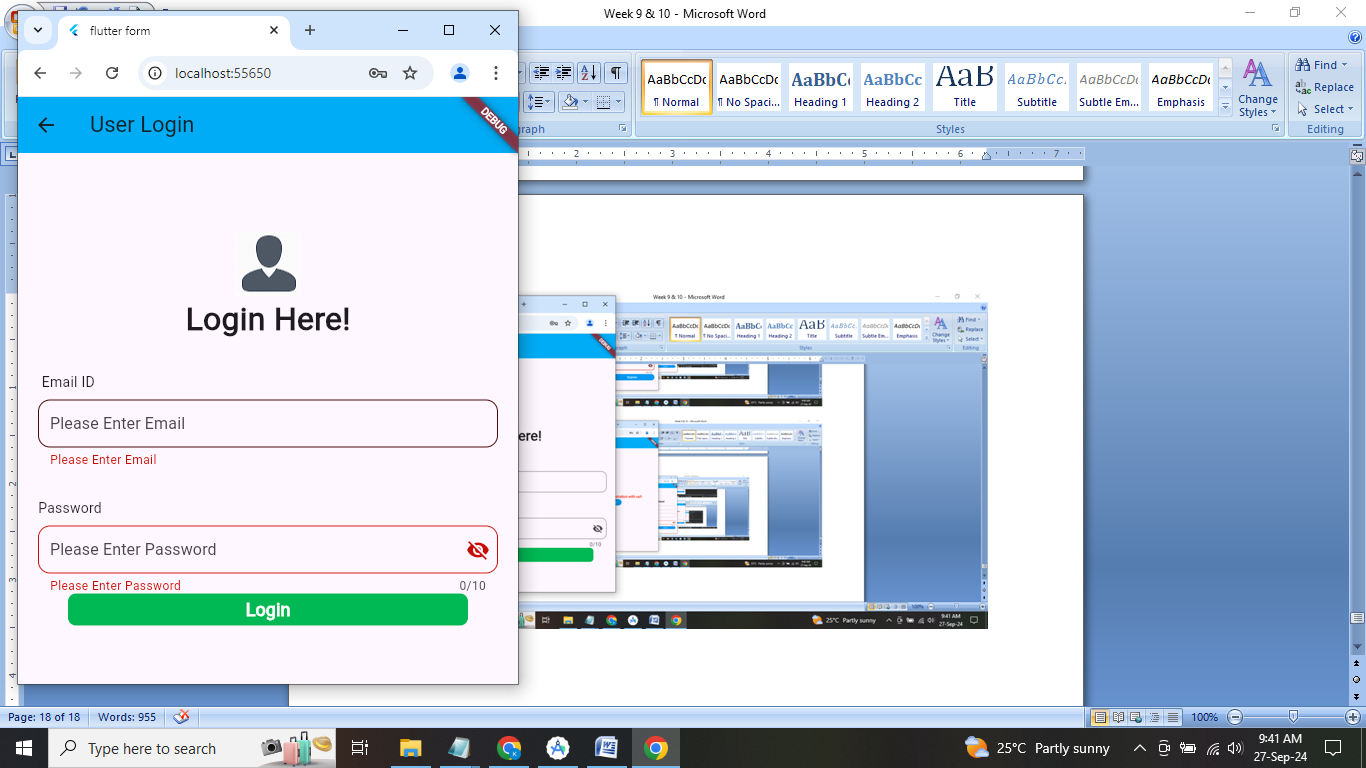
****

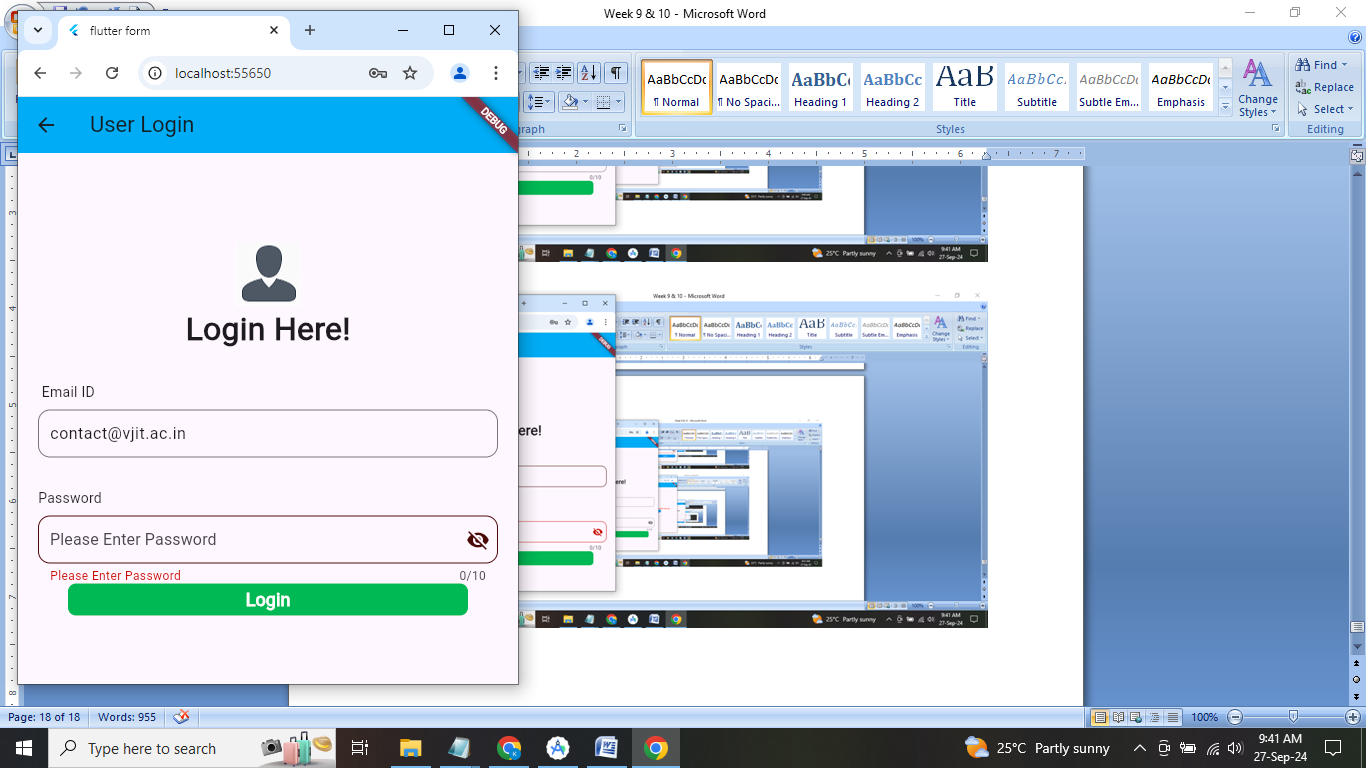
****

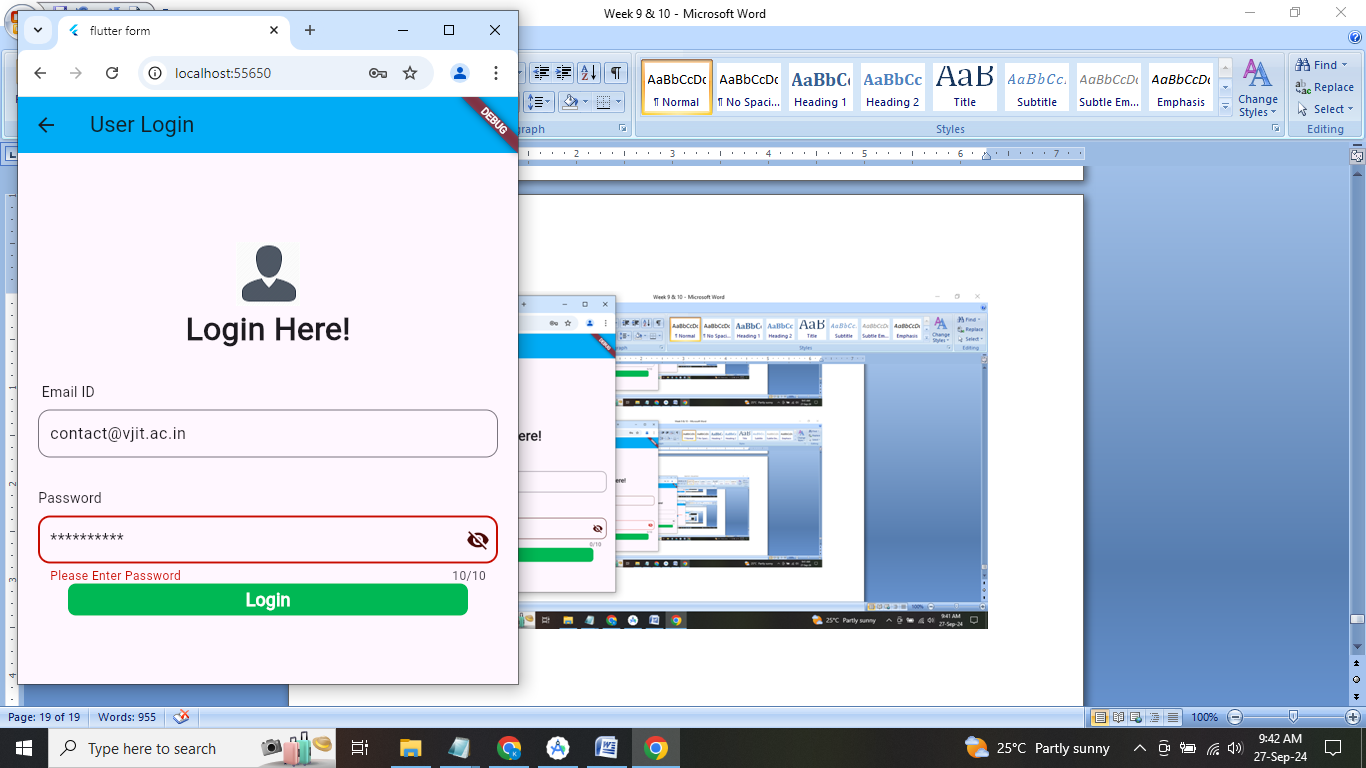
****

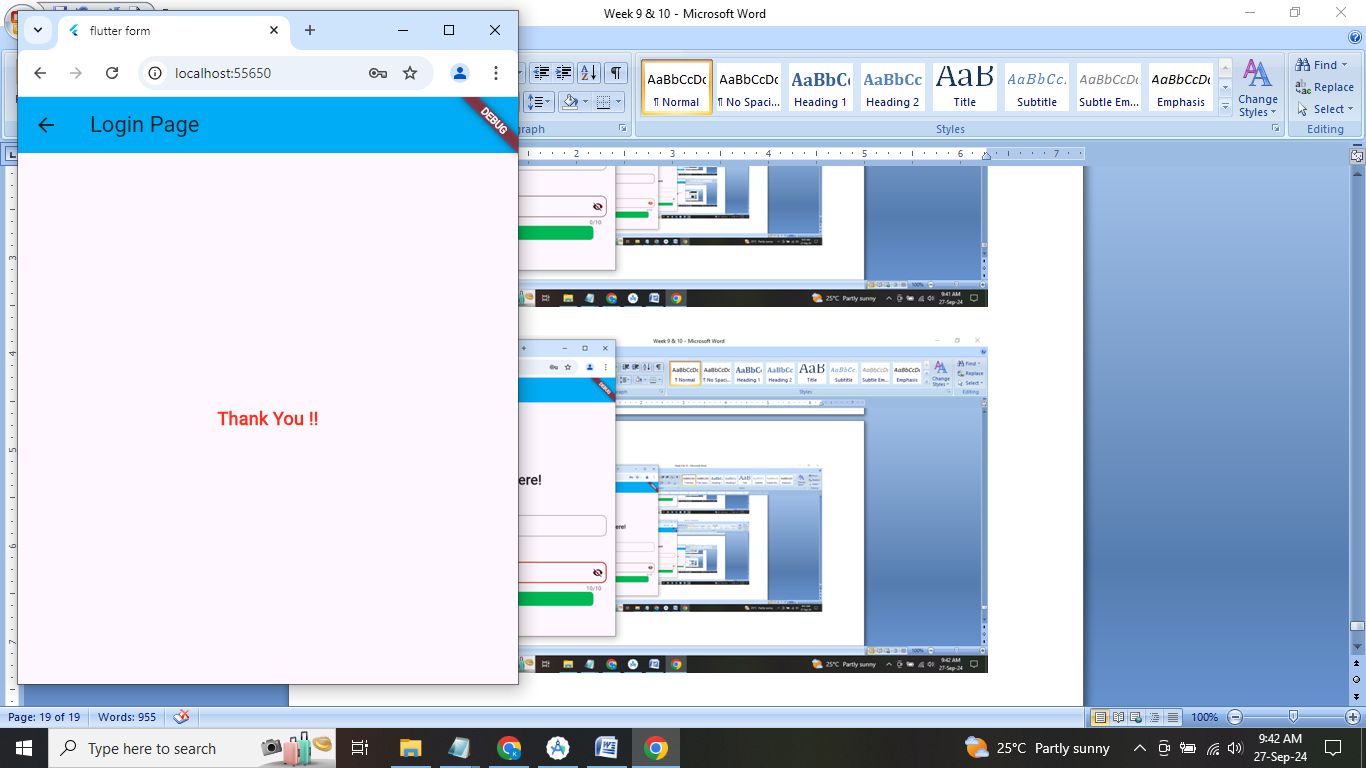
****

****

****

****

****

****