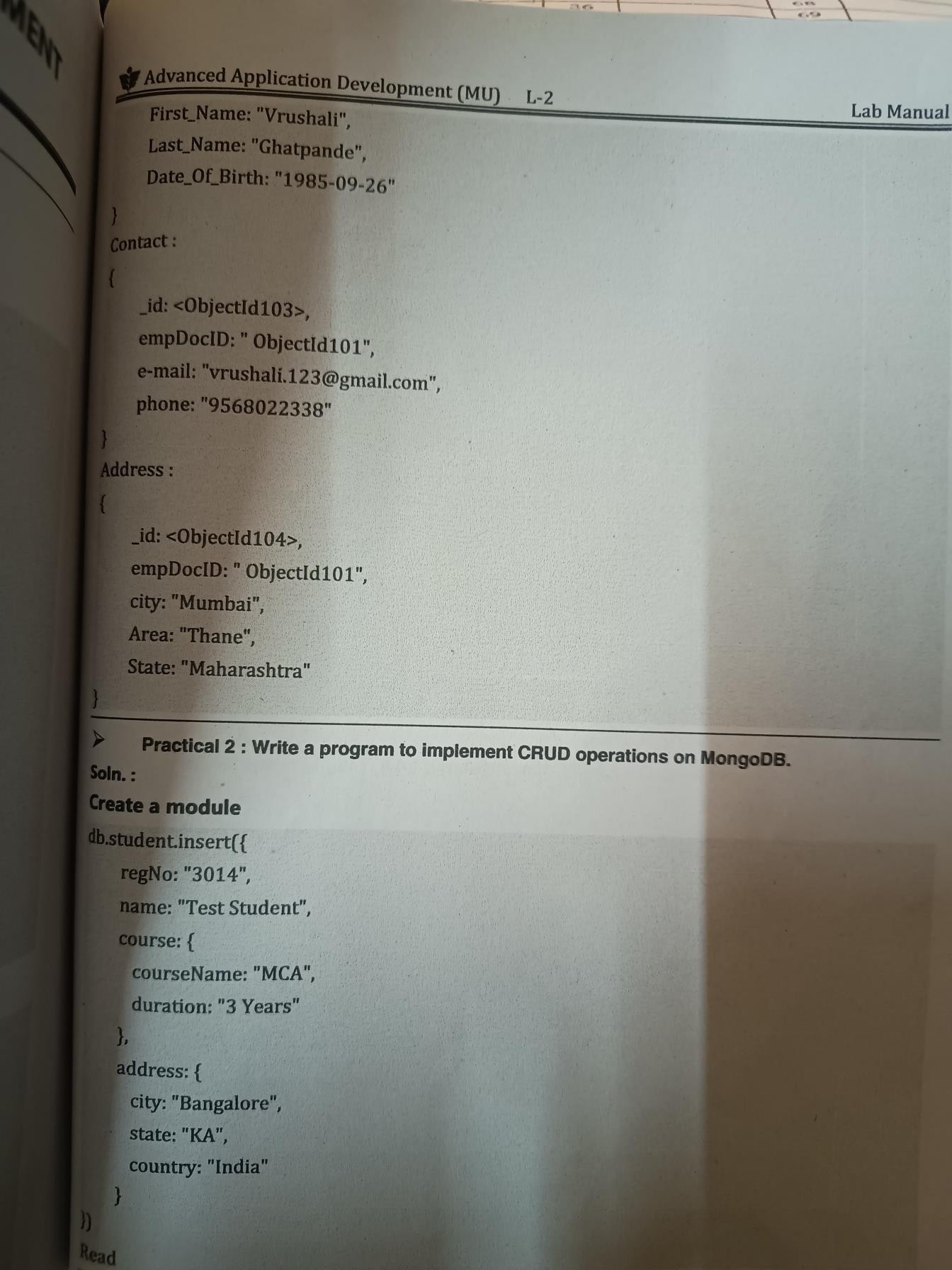
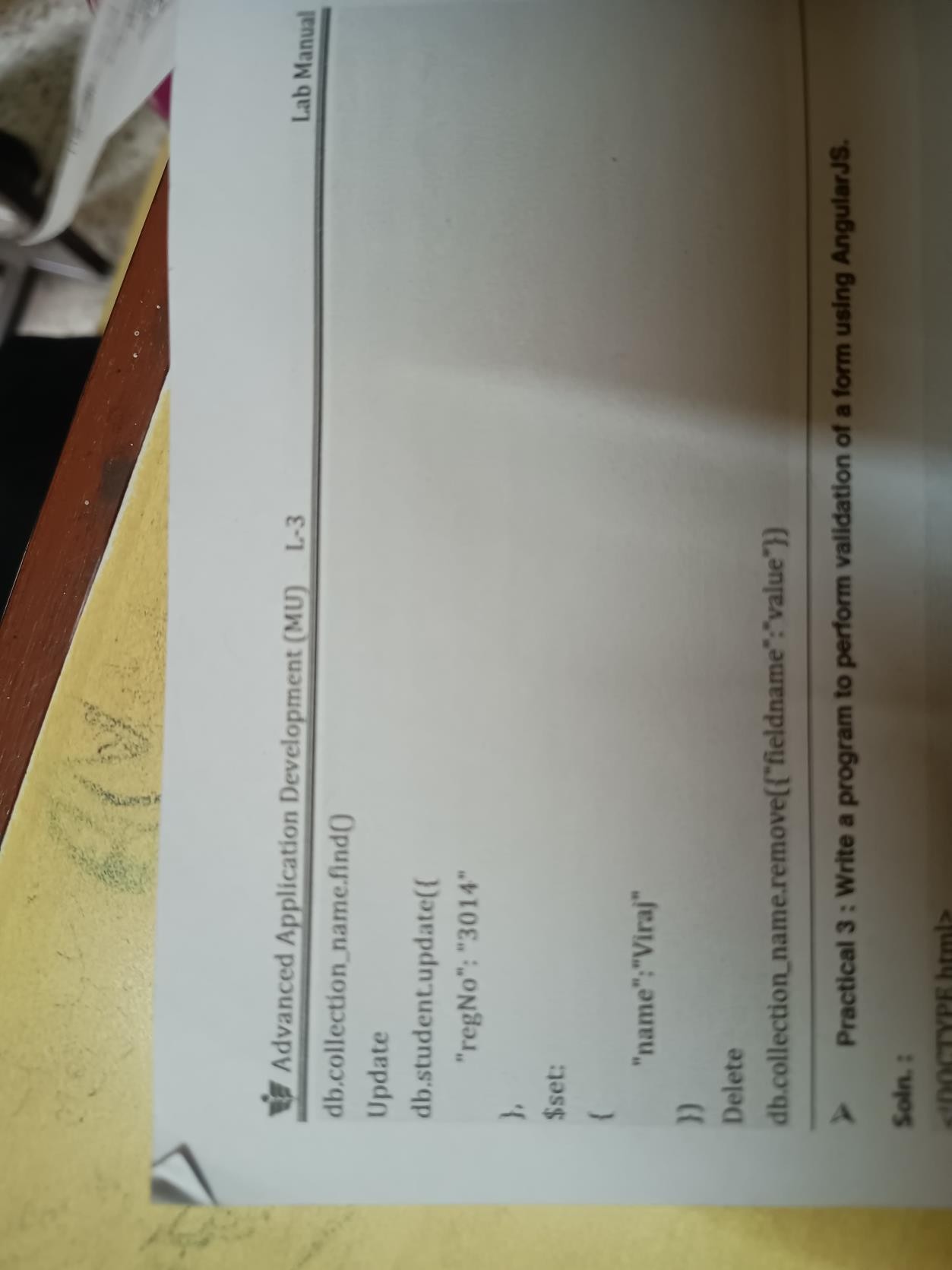
pg. 1



pg. 2

pg. 3

***PRACITACL NO:3***

### Aim : Write a program to perform validation of form using angular js.

Program:

<!DOCTYPE html>

<html>

<head>

<title>AngularJS Form Validation</title>

<script src=

"https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">

</script>

<style>

body {

text-align: center;

font-family: Arial, Helvetica, sans-serif;

}

h1 {

color: green;

}

</style>

</head>

<body ng-app="developapp">

pg. 4

<h1>GeeksforGeeks</h1>

<h3>AngularJS Custom Form Validation</h3>

<form name="form1"> Username:

<input name="username" required><br><br> Age:

<input name="userage"

ng-model="userage" required app-directive>

</form>

<p>The input's valid state is:</p>

<h3>{{form1.userage.$valid}}</h3>

<script>

var app = angular.module('developapp', []); app.directive('appDirective', function () {

return {

require: 'ngModel',

link: function (scope, element, attr, mCtrl) { function myValidation(value) {

if (value >= 18) {

mCtrl.$setValidity('charE', true);

} else {

pg. 5

mCtrl.$setValidity('charE', false);

}

return value;

}

};

});

}

mCtrl.$parsers.push(myValidation);

</script>

<p>

<b>Note:</b>

The input field must have user age greater than 18 to be considered valid for voting.

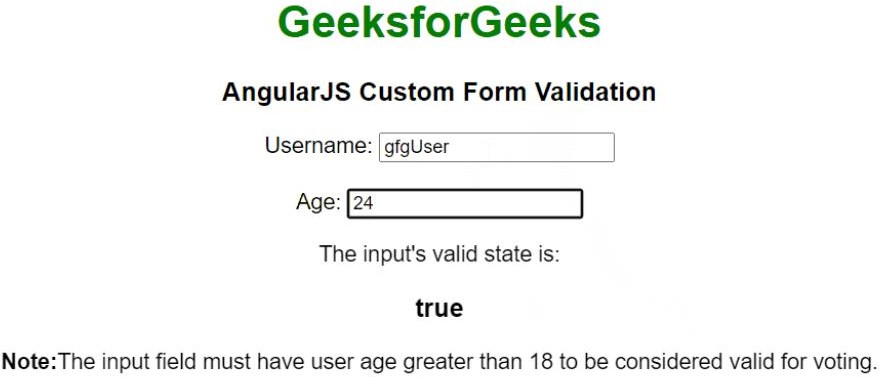
</p>

</body>

</html>

pg. 6

Output:



pg. 7

# Practical no.4

### Aim : Write a program to create and implement modules and controllers in angular js.

* Creating a Module

<div ng-app="myApp">...</div>

<script>

var app = angular.module("myApp", []);

</script>

* Adding a Controller

<!DOCTYPE html>

<html>

<script

src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></sc ript>

<body>

<div ng-app="myApp" ng-controller="myCtrl">

{{ firstName + " " + lastName }}

</div>

pg. 8

<script>

var app = angular.module("myApp", []); app.controller("myCtrl", function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

});

</script>

</body>

</html>

**Output:**



pg. 9

* Modules And Controllers In File

<!DOCTYPE html>

<html>

<script

src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></sc ript>

<body>

<div ng-app="myApp" ng-controller="myCtrl">

{{ firstName + " " + lastName }}

</div>

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

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

</body>

</html>

**Output :**

hghdachkjdgcjckd

pg. 10

**PRACITACL NO:5**

Aim : Write a program to implement Error Handling in Angular JS

<!DOCTYPE html>

<html>

<head>

<script src=

"https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js">

</script>

<style>

h1 {

color: green

}

button {

color: white;

background-color: black; height: 30px;

width: 160px; padding: 3px;

margin: 5px;

pg. 11

</style>

</head>

border-radius: 5px;

}

<body ng-app="myApp">

<center>

<h1> Geeksforgeeks</h1>

<div ng-controller="MyCtrl">

<button ng-click="generateError()"> generateError

</button>

</div>

</center>

<script>

angular.module('myApp', [])

.factory

('errorService', function ($exceptionHandler) {

return function (message) {

$exceptionHandler(message);

}

})

pg. 12

.controller('MyCtrl', function ($scope, errorService) {

$scope.generateError = function () {

errorService("Error: This is a sample custom

Error!");

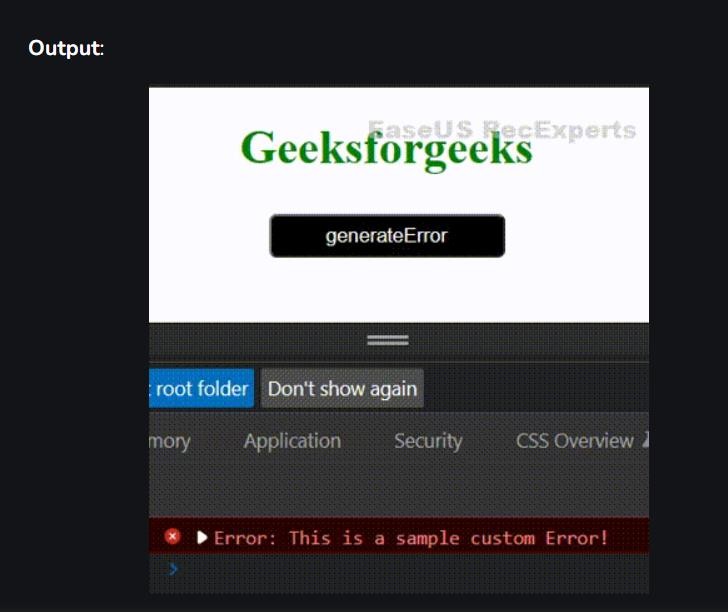
}

</script>

});

</body>

</html>



pg. 13

# Practical no.6

Aim : Create an application for Customer / Students records using AngularJS

<!DOCTYPE html>

<html>

<head>

<title>

Angular show hide table element on click

</title>

<script src=

"https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

</script>

</head>

<body>

<div ng-app="mainApp" ng-controller="MyCtrl">

<button ng-click="showPresent()"> Show People Present

</button>

<button ng-click="showAbsent()"> Show People Absent

pg. 14

</button>

<table>

<tr>

</tr>

<td>Name</td>

<td>Age</td>

<!-- The next two lines are interchangeable-->

<!-- <tr ng-repeat="p in people" ng- hide="p.attended!=flag"></tr>-->

<tr ng-repeat="p in people" ng-show="p.attended==flag">

<td>{{p.name}}</td>

<td>{{p.Age}}</td>

</tr>

</table>

</div>

<script>

var app = angular.module("mainApp", []);

app.controller("MyCtrl", function($scope) {

$scope.flag = 1;

$scope.people = [{

name: "GeekAgashi", Age: 12,

attended: 1

pg. 15

}, {

}];

}, {

name: "GeekSatoshi", Age: 16,

attended: 0

name: "GeekNakumato", Age: 14,

attended: 1

$scope.showPresent = function() {

$scope.flag = 1;

};

$scope.showAbsent = function() {

$scope.flag = 0;

};

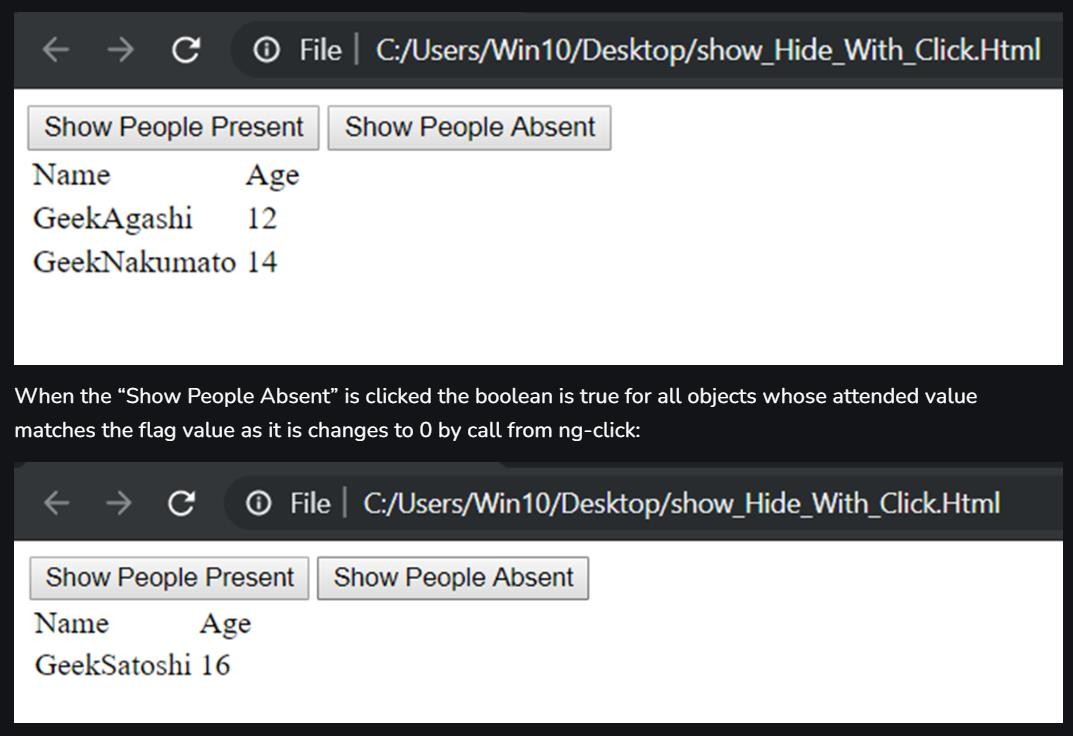
});

</script>

</body>

</html>

pg. 16



pg. 17

# Practical no.7

Aim : Write a program to create a simple web application using Express, Node JS and Angular JS

<!DOCTYPE html>

<html>

<link rel="stylesheet" href="/w3css/4/w3.css">

<script

src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.mi n.js"></script>

<body ng-app="myApp" ng-controller="userCtrl">

<div class="w3-container">

<h3>Users</h3>

<table class="w3-table w3-bordered w3-striped">

<tr>

<th>Edit</th>

<th>First Name</th>

<th>Last Name</th>

</tr>

<tr ng-repeat="user in users">

pg. 18

<td>

<button class="w3-btn w3-ripple" ng-

click="editUser(user.id)">&#9998; Edit</button>

</td>

<td>{{ user.fName }}</td>

<td>{{ user.lName }}</td>

</tr>

</table>

<br>

<button class="w3-btn w3-green w3-ripple" ng-

click="editUser('new')">&#9998; Create New User</button>

<form ng-hide="hideform">

<h3 ng-show="edit">Create New User:</h3>

<h3 ng-hide="edit">Edit User:</h3>

<label>First Name:</label>

<input class="w3-input w3-border" type="text" ng-

model="fName" ng-disabled="!edit" placeholder="First Name">

<br>

<label>Last Name:</label>

pg. 19

<input class="w3-input w3-border" type="text" ng-model="lName" ng-disabled="!edit" placeholder="Last Name">

<br>

<label>Password:</label>

<input class="w3-input w3-border" type="password" ng- model="passw1" placeholder="Password">

<br>

<label>Repeat:</label>

<input class="w3-input w3-border" type="password" ng- model="passw2" placeholder="Repeat Password">

<br>

<button class="w3-btn w3-green w3-ripple" ng-disabled="error || incomplete">&#10004; Save Changes</button>

</form>

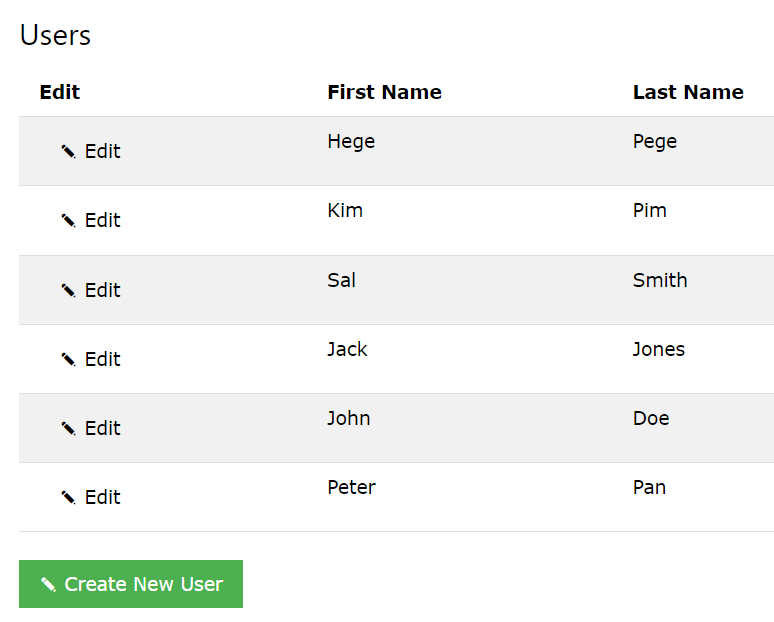
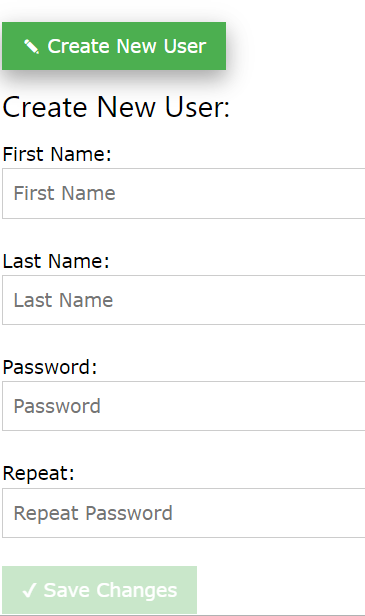
</div>

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

</body>

</html>

pg. 20



pg. 21

# Practical no.8

Aim : Create a simple HTML “Hello World” Project using AngularJS Framework and apply ng-controller, ng-model and expressions

<!DOCTYPE html>

<html>

<script

src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></sc ript>

<body>

<div ng-app="myApp" ng-controller="personCtrl">

First Name: <input type="text" ng-model="firstName"><br> Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{fullName()}}

</div>

<script>

var app = angular.module('myApp', []); app.controller('personCtrl', function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

pg. 22

$scope.fullName = function() {

return $scope.firstName + " " + $scope.lastName;

};

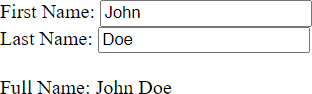
});

</script>

</body>

</html>

## Output:



pg. 23

# Practical no.9

Aim : Create an app using Flutter for User Authentication

import 'package:flutter/material.dart'; void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget { const MyApp({super.key});

@override

Widget build(BuildContext context) { return MaterialApp(

debugShowCheckedModeBanner: false, title: 'Geeks For Geeks Passing Data',

theme: ThemeData(

primarySwatch: Colors.green,

),

PassDataDemo(),

);

pg. 24

}

}

class PassDataDemo extends StatefulWidget { const PassDataDemo({Key? key}) : super(key: key); @override

State<PassDataDemo> createState() => \_PassDataDemoState();

}

class \_PassDataDemoState extends State<PassDataDemo> { final myController = TextEditingController();

@override void dispose() {

myController.dispose(); super.dispose();

}

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

pg. 25

title: Text("Geeks For Geeks"),

),

body: Center(

child: Column( children: [

SizedBox( height: 50,

),

Container( height: 50,

width: 300,

margin: EdgeInsets.all(8), decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

border: Border.all(color: Colors.black, width: 1),

),

child: TextField(

controller: myController,

decoration: InputDecoration(

pg. 26

labelText: 'Enter The Data', border: InputBorder.none,

floatingLabelBehavior: FloatingLabelBehavior.never,

),

),

),

SizedBox( height: 30,

),

// Step 1 <-- SEE HERE

ElevatedButton( onPressed: () {

// Step 3 <-- SEE HERE

Navigator.push( context,

MaterialPageRoute(

builder: (context) =>

DetailScreen(title: myController.text),

),

pg. 27

);

},

child: const Text(

'Pass Data To Next Screen', style: TextStyle(fontSize: 24),

),

),

],

),

),

);

}

}

class DetailScreen extends StatefulWidget {

const DetailScreen({Key? key, required this.title}) : super(key: key); final String title;

@override

State<DetailScreen> createState() => \_DetailScreenState();

}

pg. 28

class \_DetailScreenState extends State<DetailScreen> { @override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: Text("Screen Two"),

),

body: Center(

child: Column( children: [

SizedBox( height: 50,

),

Text(

'${widget.title}',

style: TextStyle(fontSize: 54),

),

pg. 29

ElevatedButton( onPressed: () {

Navigator.pop(context);

},

child: Text("Write New Text"),

)

],

),

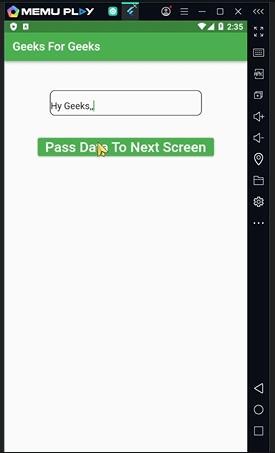
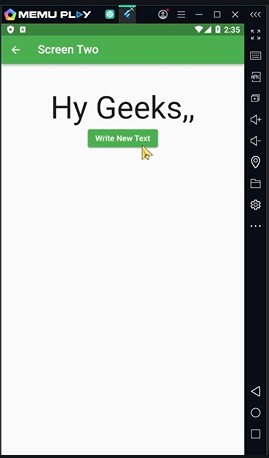
),

);

}

}

pg. 30

pg. 31