

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

- 1. Write an AngularJS script for addition of two numbers using ng-init, ng-model & ng bind. And also demonstrate ng-show, ng-disabled, ng-click directives on button component.**

```
<!DOCTYPE html>
<html ng-app="AdditionApp">
<head>
  <title>AngularJS Addition Example</title>
  <script
    src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="AdditionController" ng-init="num1=0; num2=0; result=0; showResult=false">

  <h2>Simple Addition Calculator</h2>

  <label>Number 1:</label>
  <input type="number" ng-model="num1" />

  <br><br>

  <label>Number 2:</label>
  <input type="number" ng-model="num2" />

  <br><br>

  <button
    ng-click="addNumbers()"
    ng-disabled="num1 === null || num2 === null"
  >
    Add
  </button>

  <br><br>

  <div ng-show="showResult">
    <strong>Result:</strong> <span ng-bind="result"></span>
  </div>

  <script>
    angular.module('AdditionApp', [])
      .controller('AdditionController', function($scope) {
        $scope.addNumbers = function() {
```

```

$scope.result = $scope.num1 + $scope.num2;
$scope.showResult = true;
        Kannada Sangha Pune's
        Kaveri College of Arts, Science and Commerce
        Fullstack Development II

```

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```

    } ;
    }) ;
</script>

</body>
</html>

```

2. Create a Node.js application that reads data from multiple files asynchronously using promises and async/await

```

const fs = require('fs').promises;
const path = require('path');

const files = ['file1.txt', 'file2.txt', 'file3.txt'];
const dataDir = path.join(__dirname, 'data'); // files under data directory

async function readFiles(fileList) {
  try {
    const readPromises = fileList.map(file =>
      fs.readFile(path.join(dataDir, file), 'utf-8') );
    const contents = await Promise.all(readPromises);

    contents.forEach((content, index) => {
      console.log(` ${fileList[index]}:\n${content}\n`); });
    } catch (error) {
      console.error(' Error reading files:', error.message);
    }
}

readFiles(files);

```

3. Write an AngularJS script to print details of bank (bank name, MICR code, IFC code, address etc.) in tabular form using ng-repeat.

```

<!DOCTYPE html>
<html ng-app="BankApp">
<head>
  <title>Bank Details Table</title>
  <script
  src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

```

```
</head>
```

Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
<body ng-controller="BankController">

<h2>Bank Details</h2>

<table>
<thead>
<tr>
<th>Bank Name</th>
<th>MICR Code</th>
<th>IFSC Code</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr ng-repeat="bank in banks">
<td>{{ bank.name }}</td>
<td>{{ bank.micr }}</td>
<td>{{ bank.ifsc }}</td>
<td>{{ bank.address }}</td>
</tr>
</tbody>
</table>

<script>
angular.module('BankApp', [])
.controller('BankController', function($scope) {
$scope.banks = [
{
name: 'State Bank of India',
micr: '400002002',
ifsc: 'SBIN0000300',
address: 'Mumbai Main Branch, Mumbai' },
{
name: 'HDFC Bank',
micr: '110240001',
ifsc: 'HDFC0000001',
address: 'Connaught Place, New Delhi' },
{
name: 'ICICI Bank',
micr: '560229002',
ifsc: 'ICIC0000031',
address: 'MG Road, Bengaluru'
}
```

```
];
});
```

Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
</script>

</body>
</html>
```

4. **Create a simple Angular application that fetches data from an API using HttpClient. Implement an Observable to fetch data from an API endpoint.**

Go through Practical 1 PPT

5. **Write an AngularJS script to display list of games stored in an array on click of button using ng-click and also demonstrate ng-init, ng-bind directive of AngularJS.**

```
<!DOCTYPE html>
<html ng-app="GameApp">
<head>
  <title>Game List</title>
  <script
    src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="GameController" ng-init="games=[]; showGames=false; heading='List of Games'>

  <h2 ng-bind="heading"></h2>

  <button ng-click="loadGames()">Show Games</button>

  <ul ng-show="showGames">
    <li ng-repeat="game in games">{{ game }}</li>
  </ul>

  <script>
    angular.module('GameApp', [])
      .controller('GameController', function($scope) {
        $scope.loadGames = function() {
          $scope.games = [
            'Chess',
            'Football',
            'Cricket',
            'Badminton',
            'Hockey',
            'Tennis'
          ]
        }
      })
  </script>
```

```
];
$scope.showGames = true;
};
}) ;
```

Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
</script>

</body>
</html>
```

6. Find a company with a workforce greater than 30 in the array (use find by id method) // Define an array of company objects

```
const companies = [
  { id: 1, name: 'TechNova', workforce: 25 },
  { id: 2, name: 'GreenSoft', workforce: 40 },
  { id: 3, name: 'CloudCore', workforce: 18 },
  { id: 4, name: 'DataForge', workforce: 55 }
];

// Use find() to locate the first company with workforce > 30
const company = companies.find(c => c.workforce > 30);

// Display the result
if (company) {
  console.log('Company with workforce > 30 found:');
  console.log(`ID: ${company.id}`);
  console.log(`Name: ${company.name}`);
  console.log(`Workforce: ${company.workforce}`);
} else {
  console.log('No company found with workforce greater than 30.');
}
```

7. Fetch the details using ng-repeat in AngularJS

```
<!DOCTYPE html>
<html ng-app="StudentApp">
<head>
  <title>Student Grades</title>
  <script
    src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="StudentController" ng-init="students=[];
showTable=false; heading='Student Grade Report'">
```

```

<h2 ng-bind="heading"></h2>

<button ng-click="loadStudents () ">Show Grades</button>

<table border="1" ng-show="showTable">
<thead>
    Kannada Sangha Pune's
    Kaveri College of Arts, Science and Commerce
    Fullstack Development II

    Assignment solutions

```

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```

<tr>
<th>Name</th>
<th>Subject</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr ng-repeat="student in students">
<td>{{ student.name }}</td>
<td>{{ student.subject }}</td>
<td>{{ student.grade }}</td>
</tr>
</tbody>
</table>

<script>
angular.module('StudentApp', [])
.controller('StudentController', function($scope) {
$scope.loadStudents = function() {
$scope.students = [
{ name: 'Aarav', subject: 'Math', grade: 'A' }, { name:
'Isha', subject: 'Science', grade: 'B+' }, { name: 'Rohan',
subject: 'English', grade: 'A-' }, { name: 'Sneha', subject:
'History', grade: 'B' } ];
$scope.showTable = true;
};
});
</script>

</body>
</html>

```

8. Express.js application to include middleware for parsing request bodies (e.g., JSON, form data) and validating input data.

```
npm install express body-parser express-validator
```

```
// HTML File
<form action="http://localhost:3000/submit"
method="POST"> <input type="text" name="name"
placeholder="Name" /> <input type="email" name="email"
placeholder="Email" /> <input type="number" name="age"
placeholder="Age" /> <button
type="submit">Submit</button>
</form>
```

Kannada Sangha Pune's
 Kaveri College of Arts, Science and Commerce
 Fullstack Development II

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
// node js file
const express = require('express');
const bodyParser = require('body-parser');
const { body, validationResult } = require('express-validator');

const app = express();
const PORT = 3000;

// Middleware to parse JSON and form data
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// Sample POST route with validation
app.post('/submit',
  [
    body('name').isLength({ min: 2 }).withMessage('Name must be at least 2 characters'),
    body('email').isEmail().withMessage('Invalid email address'),
    body('age').isInt({ min: 1 }).withMessage('Age must be a positive integer')
  ],
  (req, res) => {
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({ errors: errors.array() });
    }

    const { name, email, age } = req.body;
    res.json({ message: 'Data received successfully', data: { name, email, age } });
  }
);

// Start server
```

```

app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});

```

9. Create a simple Angular component that takes input data and displays it.

ng generate component user-input

```

// user-input.component.ts
import { Component } from '@angular/core';
Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

```

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```

@Component({
  selector: 'app-user-input',
  templateUrl: './user-input.component.html'
})
export class UserInputComponent {
  userName: string = '';
}

//user-input.component.html
<h2>Enter Your Name</h2>

<input type="text" [(ngModel)]="userName" placeholder="Your name" />

<p *ngIf="userName">Hello, {{ userName }}!</p>

//Enable FormsModule in app.module.ts
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';

import { AppComponent } from './app.component';
import { UserInputComponent } from './user-input/user-input.component';

@NgModule({
  declarations: [AppComponent, UserInputComponent],
  imports: [BrowserModule, FormsModule],
  bootstrap: [AppComponent]
})
export class AppModule {}

//Use Component in app.component.html
<app-user-input></app-user-input>

```

```
ng serve
```

10.Implement a simple server using Node.js.

```
const http = require('http');

// Create the server
const server = http.createServer((req, res) => {
  // Set response header
  res.writeHead(200, { 'Content-Type': 'text/plain' });

  // Send response

```

Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
res.end('Hello! This is a simple Node.js server.');
});

// Start listening on port 3000
const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Server is running at http://localhost:${PORT}`);
});
```

11.Develop an Express.js application that defines routes for Create and Read operations on a resource (products).

Change the same code for users, employees, movies. Instead of product everything will be for users, employees, movies.

```
npm install express body-parser
```

```
//express code
const express = require('express');
const bodyParser = require('body-parser');

const app = express();
const PORT = 3000;

// Middleware to parse JSON request bodies
app.use(bodyParser.json());

// In-memory product store
let products = [];

// CREATE: Add a new product
app.post('/products', (req, res) => {
```

```

const { name, price } = req.body;

if (!name || !price) {
  return res.status(400).json({ error: 'Name and price are required.' });
}

const newProduct = {
  id: products.length + 1,
  name,
  price
};

products.push(newProduct);

```

**Kannada Sangha Pune's
 Kaveri College of Arts, Science and Commerce
 Fullstack Development II**

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```

res.status(201).json({ message: 'Product created', product:
newProduct });
});

// READ: Get all products
app.get('/products', (req, res) => {
  res.json(products);
});

// READ: Get a product by ID
app.get('/products/:id', (req, res) => {
  const productId = parseInt(req.params.id);
  const product = products.find(p => p.id === productId);

  if (!product) {
    return res.status(404).json({ error: 'Product not found.' });
  }
  res.json(product);
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});

```

Following are the routes and methods which will work for the given purpose

Route	Method	Purpose
-------	--------	---------

/products	POST	Create a new product
/products	GET	Retrieve all products
/products/:id	GET	Retrieve a product by its ID

Use either curl command or postman to execute different routes

12.Create Express.js application to include middleware for parsing request bodies (e.g., JSON, form data) and validating input data. Send appropriate JSON responses for success and error cases.

```
npm install express body-parser express-validator
```

```
// Express code
const express = require('express');
const bodyParser = require('body-parser');
const { body, validationResult } = require('express-validator');
```

Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
const app = express();
const PORT = 3000;

// Middleware to parse JSON and form data
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// POST /register - Validate and process user input
app.post('/register',
  [
    body('username').isLength({ min: 3 }).withMessage('Username must be at least 3 characters'),
    body('email').isEmail().withMessage('Invalid email address'),
    body('age').isInt({ min: 1 }).withMessage('Age must be a positive number')
  ],
  (req, res) => {
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({
        success: false,
        errors: errors.array()
      });
    }
  }
);
```

```

    });

    const { username, email, age } = req.body;
    res.status(200).json({
      success: true,
      message: 'User registered successfully',
      data: { username, email, age }
    });
  }
);

// Start the server
app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});

```

13.Express.js application to include middleware for parsing request bodies (e.g., JSON, form data) and validating input data. Send appropriate JSON responses for success and error cases.

npm install express body-parser express-validator
 Kannada Sangha Pune's
 Kaveri College of Arts, Science and Commerce
 Fullstack Development II

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```

const express = require('express');
const bodyParser = require('body-parser');
const { body, validationResult } = require('express-validator');

const app = express();
const PORT = 3000;

// Middleware to parse JSON and form data
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// POST /submit - Validate and process input
app.post('/submit',
  [
    body('username').isLength({ min: 3 }).withMessage('Username must be at least 3 characters'),
    body('email').isEmail().withMessage('Invalid email address'),
    body('age').isInt({ min: 1 }).withMessage('Age must be a positive number')
  ],
  (req, res) => {

```

```

const errors = validationResult(req);
if (!errors.isEmpty()) {
  return res.status(400).json({
    success: false,
    errors: errors.array()
  });
}

const { username, email, age } = req.body;
res.status(200).json({
  success: true,
  message: 'Data received successfully',
  data: { username, email, age }
});
}
);

// Start the server
app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});

```

Kannada Sangha Pune's
Kaveri College of Arts, Science and Commerce
Fullstack Development II

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

14.Create Angular application that print the name of students who play basketball using filter and map method.

ng generate component basketball-players

```
//basketball-players.component.ts
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-basketball-players',
  templateUrl: './basketball-players.component.html'
})
export class BasketballPlayersComponent implements OnInit
{  students = [
    { name: 'Aarav', sport: 'cricket' },
    { name: 'Isha', sport: 'basketball' },
    { name: 'Rohan', sport: 'football' },
    { name: 'Sneha', sport: 'basketball' },
    { name: 'Kabir', sport: 'tennis' }
  ];

  basketballPlayers: string[] = [];

  ngOnInit()  {

```

```

this.basketballPlayers = this.students
  .filter(student => student.sport === 'basketball')
  .map(student => student.name);
}

//basketball-players.component.html
<h2>Basketball Players</h2>
<ul>
  <li *ngFor="let player of basketballPlayers">{{ player
}}</li>
</ul>

//Add Component to app.component.html
<app-basketball-players></app-basketball-players>

```

15. Write an AngularJS script to print details of Employee (employee name, employee Id, Pin code, address etc.) in tabular form using ng-repeat.

```

<!DOCTYPE html>
<html ng-app="EmployeeApp">
<head>
  <title>Employee Details</title>
    Kannada Sangha Pune's
    Kaveri College of Arts, Science and Commerce
    Fullstack Development II

```

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```

<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="EmployeeController" ng-init="loadEmployees()">

<h2>Employee Directory</h2>

<table border="1" cellpadding="8" cellspacing="0">
<thead>
  <tr>
    <th>Employee Name</th>
    <th>Employee ID</th>
    <th>Pin Code</th>
    <th>Address</th>
  </tr>
</thead>
<tbody>
  <tr ng-repeat="emp in employees">
    <td>{{ emp.name }}</td>

```

```

<td>{{ emp.id }}</td>
<td>{{ emp.pin }}</td>
<td>{{ emp.address }}</td>
</tr>
</tbody>
</table>

<script>
angular.module('EmployeeApp', [])
.controller('EmployeeController', function($scope) {
$scope.loadEmployees = function() {
$scope.employees = [
{ name: 'Amit Sharma', id: 'E101', pin: '411001', address: 'Pune, Maharashtra' },
{ name: 'Neha Verma', id: 'E102', pin: '110011', address: 'Delhi, India' },
{ name: 'Ravi Kumar', id: 'E103', pin: '560001', address: 'Bangalore, Karnataka' },
{ name: 'Sneha Joshi', id: 'E104', pin: '400001', address: 'Mumbai, Maharashtra' }
];
};
});
</script>

</body>

```

Kannada Sangha Pune's
 Kaveri College of Arts, Science and Commerce
 Fullstack Development II

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

```
</html>
```

Kannada Sangha Pune's
 Kaveri College of Arts, Science and Commerce
 Fullstack Development II

Assignment solutions

Wherever it says "Write an AngularJS" code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

16.Create Angular application that print the name of students who got 85% using filter and map method.

```

ng generate component top-scorers

//top-scorers.component.ts
import { Component, OnInit } from '@angular/core';

@Component({

```

```

        selector: 'app-top-scorers',
        templateUrl: './top-scorers.component.html'
    })
export class TopScorersComponent implements OnInit
{   students = [
    { name: 'Aarav', percentage: 78 },
    { name: 'Isha', percentage: 85 },
    { name: 'Rohan', percentage: 92 },
    { name: 'Sneha', percentage: 85 },
    { name: 'Kabir', percentage: 80 }
];

topScorers: string[] = [];

ngOnInit() {
this.topScorers = this.students
.filter(student => student.percentage === 85)
.map(student => student.name);
}
}

//top-scorers.component.html
<h2>Students Who Scored 85%</h2>
<ul>
<li *ngFor="let name of topScorers">{{ name }}</li>
</ul>

//Use Component in app.component.html
<app-top-scorers></app-top-scorers>

```

Explanation of functions

Method	Purpose
filter()	Selects students with exactly
map()	85% Extracts their names
*ngFor	Displays the names in a list

Kannada Sangha Pune's
 Kaveri College of Arts, Science and Commerce
 Fullstack Development II

Assignment solutions

Wherever it says “Write an AngularJS” code just create single HTML file having script tag to AngularJS schema. Open HTML file in browser and check its working.

17. Find an emp with a Salary greater than 25000 in the array. (Using find by id method) // Define an array of employee objects

```

const employees = [
  { id: 101, name: 'Amit', salary: 24000 },

```

```
{ id: 102, name: 'Neha', salary: 26000 },
{ id: 103, name: 'Ravi', salary: 28000 },
{ id: 104, name: 'Sneha', salary: 22000 }
];

// Find the first employee with salary > 25000
const highEarner = employees.find(emp => emp.salary > 25000);

// Display the result
if (highEarner) {
  console.log(`Employee Found:
Name: ${highEarner.name}
ID: ${highEarner.id}
Salary: ₹${highEarner.salary}`);
} else {
  console.log('No employee found with salary greater than
₹25,000.');
}
```