**Slip 14**

**Q1 Create Angular application that print the name of students who got 85%**

**using filter and map method**

let students = [

{ id: "001", name: "Anish",Percentage: "40%" },

{ id: "002", name: "Smriti", Percentage: "25%" },

{ id: "003", name: "Rahul", Percentage: "85%" },

{ id: "004", name: "Bakul", Percentage: "96%" },

{ id: "005", name: "Nikita", Percentage: "90%" }

]

let basketballPlayers = students.filter(function (student) {

return student.Percentage === "85%";

}).map(function (student) {

return student.name;

})

console.log("Basketball Players are:");

// Printing out the name of Basketball players

basketballPlayers.forEach(function (players) {

console.log(players);

});

**Q2 Develop an Express.js application that defines routes for Create, Update**

**operations on a resource (Employee).**

const express = require('express');

const bodyParser = require('body-parser');

const app = express();

const PORT = 3000;

// Middleware for parsing JSON bodies

app.use(bodyParser.json());

// In-memory "database" to store users

let users = [];

// Create a new user (Create)

app.post('/emp', (req, res) => {

const { name, email } = req.body;

const newEmp= { id:emp.length + 1, name, email };

emp.push(newEmp);

res.status(201).json(newEmp);

});

// Read all users (Read)

app.get('/emp', (req, res) => {

res.json(emp);

});

// Update a user by ID (Update)

app.put('/emp /:id', (req, res) => {

const userId = parseInt(req.params.id, 10);

const userIndex = users.findIndex(u => u.id === userId);

if (userIndex === -1) {

return res.status(404).json({ message: 'User not found' });

}

const { name, email } = req.body;

emp[userIndex] = { id: userId, name, email };

res.json(users[userIndex]);

});

// Start the server

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});