**Slip 9**

**Q1 find a company with a workforce greater than 30 in the array. used find by**

**id method**

interface Company {

name: string;

desc: string;

workForce: number;

}

const companies: Company[] = [

{ name: "GeeksforGeeks", desc: "A Computer Science Portal.", workForce: 200 },

{ name: "Company 2", desc: "Description 1", workForce: 30 },

{ name: "Company 3", desc: "Description 2", workForce: 10 },

];

const matchedCompany = companies.find(company => company.workForce > 30);

console.log(matchedCompany);

**Q2 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.**

const express = require('express');

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

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

const app = express();

const PORT = 3000;

// Middleware for parsing JSON and form data

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

// Sample route with input validation

app.post(

'/submit',

[

// Validation checks

body('username').isLength({ min: 3 }).withMessage('Username must be at least 3

characters long'),

body('email').isEmail().withMessage('Enter a valid email address'),

body('age').isInt({ min: 1 }).withMessage('Age must be a positive integer'),

],

(req, res) => {

// Check validation results

const errors = validationResult(req);

if (!errors.isEmpty()) {

return res.status(400).json({ success: false, errors: errors.array() });

}

// If validation passes, process the request

const { username, email, age } = req.body;

res.status(200).json({ success: true, message: 'Data received successfully', data: {

username, email, age } });

}

);

// Default route

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

res.send('Welcome to the Express validation application!');

});

// Start the server

app.listen(PORT, () => {

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

});