**Slip 7**

**Q1 Create a Node.js application that reads data from multiple files**

**asynchronously using**

**promises and async/await.**

const fs = require('fs');

const readFilePromise = (fileName, encoding) => {

return new Promise((resolve, reject) => {

fs.readFile(fileName, encoding, (err, data) => {

if (err) {

return reject(err);

}

resolve(data);

});

});

}

readFilePromise('./input.txt', 'utf8')

.then(data => {

console.log(data);

})

.catch(err => {

console.log(err);

});

**Q2 Develop an Express.js application that defines routes for Create and Read**

**operations**

**on a resource (User).**

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('/users', (req, res) => {

const { name, email } = req.body;

const newUser = { id: users.length + 1, name, email };

users.push(newUser);

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

});

// Read all users (Read)

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

res.json(users);

});

// Read a single user by ID (Read)

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

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

const user = users.find(u => u.id === userId);

if (!user) {

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

}

res.json(user);

});

// Start the server

app.listen(PORT, () => {

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

});