|  |  |
| --- | --- |
| Practical: 20 | App.js  *// Create CRUD Operation page using ExpressJs, MongoDB*  const express = require('express');  const bodyParser = require('body-parser');  const app = express();  *// Middleware for parsing JSON requests*  app.use(bodyParser.json());  *// Configure your routes here*  const PORT = process.env.PORT || 3000;  app.listen(PORT, () => {    console.log(`Server is running on port ${PORT}`);  });  const mongoose = require('mongoose');  mongoose.connect('mongodb://localhost:27017/dbs', {    useNewUrlParser: true,    useUnifiedTopology: true,  });  mongoose.connection.on('connected', () => {    console.log('Connected to MongoDB');  });  mongoose.connection.on('error', (err) => {    console.error('Error connecting to MongoDB:', err);  });  const crudRoutes = require('./routes/crud'); *// Importing the CRUD routes*  *// Mount CRUD routes*  app.use('/api', crudRoutes);  item.js  *// models/item.js*  const mongoose = require('mongoose');  const itemSchema = new mongoose.Schema({    name: String,    description: String,  });  const Item = mongoose.model('Item', itemSchema);  module.exports = { Item };  crud.js  const express = require('express');  const router = express.Router();  const { Item } = require('./models');  *// Create a new item*  router.post('/items', async (req, res) => {    try {      const { name, description } = req.body;      const newItem = new Item({ name, description });      await newItem.save();      res.status(201).json(newItem);    } catch (error) {      res.status(500).json({ error: 'Internal Server Error' });    }  });  *// Read all items*  router.get('/items', async (req, res) => {    try {      const items = await Item.find();      res.status(200).json(items);    } catch (error) {      res.status(500).json({ error: 'Internal Server Error' });    }  });  *// Update an item*  router.put('/items/:id', async (req, res) => {    try {      const { name, description } = req.body;      const updatedItem = await Item.findByIdAndUpdate(req.params.id, { name, description }, { new: true });      res.status(200).json(updatedItem);    } catch (error) {      res.status(500).json({ error: 'Internal Server Error' });    }  });  *// Delete an item*  router.delete('/items/:id', async (req, res) => {    try {      await Item.findByIdAndRemove(req.params.id);      res.status(204).send();    } catch (error) {      res.status(500).json({ error: 'Internal Server Error' });    }  });  module.exports = router;  Folder Structure:  Practical\_20/  ├── app.js  ├── models/  │ └── item.js  ├── routes/  │ └── crud.js  └── package.json  **Create (POST):**    **Read (GET):**    **Update (PUT):**    **Delete (DELETE):** |