Question #1: What is your availability this upcoming Thursday and Friday for a 20-minute technical .NET Core interview (via Zoom).

Ans : Yes,I am available for friday 22-09-2023 for interview infact i am available later on also.

Question #2: Write .NET Core API (or NodeJS) retrieving the list of customers from the test database's "customers" table. (DB credentials provided, below. Organize your code into a structured format, removing unnecessary files/folders, leaving only code relevant to the APIs. Test your code locally.)

Ans: i have done this in nodejs using express,bodyparser,sequelize ,sequelize-cli,mysql2

Step1-connected to the database using sequelize.

Mycode for config.json file

```
{
  "development": {
    "username": "u841345258_MVCu",
    "password": "Instient@2023",
    "database": "u841345258_MVC",
    "host": "sql348.main-hosting.eu",
    "port": 3306,
    "dialect": "mysql"
  },
  "test": {
    "username": "u841345258_MVCu",
    "password": "Instient@2023",
    "database": "u841345258_MVC",
    "host": "sql348.main-hosting.eu",
    "port": 3306,
    "dialect": "mysql"
  },
  "production": {
    "username": "u841345258_MVCu",
    "password": "Instient@2023",
    "database": "u841345258_MVC",
```

```
    "host": "sql348.main-hosting.eu",
    "port": 3306,
    "dialect": "mysql"
  }
}
```

Step 2:created a model class name customer.js and code for it is this

```
'use strict';
const {
  Model
} = require('sequelize');
module.exports = (sequelize, DataTypes) => {
  class Customer extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
    }
  }
  Customer.init({
    name: {
      type : DataTypes.STRING,
      allowNull:false
    },
    country: {
      type : DataTypes.STRING,
      allowNull:false
    }
  }, {
    sequelize,
    modelName: 'customer',
    timestamps:false
  });
  return Customer;
};
```

Step3:then made the customer repository file and customer service file code for them are

```
const {customer}=require('../models/index');
class CustomerRepository{
    async getAllCustomers(){
        try {
            const customers=await customer.findAll();
            return customers;
        } catch (error) {
            throw {error};
        }
    }
}
module.exports=CustomerRepository;
```

And for the customer service is this:

```
const CustomerRepository=require('../repository/customer-repository');
class CustomerService{
    constructor(){
        this.customerRepository=new CustomerRepository();
    }
    async getCustomers(){
        try {
            const customers=this.customerRepository.getAllCustomers();
            return customers;
        } catch (error) {
            throw {error};
        }
    }
}
module.exports=CustomerService;
```

Step 4 : Created Customer controller for making the api that code is this:

```
const CustomerService=require('../services/customer-service');
const customerService=new CustomerService();
//get request url=/customer
const getAllCustomers=async (req,res)=>{
    try {
        const cutomers=await customerService.getCustomers();
        return res.status(200).json({
            data:cutomers,
            success:true,
            message:"successfully fetch all the customers",
            err:{}
        });
    } catch (error) {
        console.log(error);
        return res.status(500).json({
            data:{},
            success:false,
            message:"failed to get all customers",
            err:error
        });
    }
}
module.exports=getAllCustomers;
```

Step 5 : find call the api using app object using express in index.js main file whose code is this

```
// import express (after npm install express)
const express = require("express");

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

const { PORT } = require("./config/serverConfig");
```

```
const getAllCustomers = require("./controller/customer-controller");
const getOrdersById = require("./controller/order-controller");

const
getCustomerOrdersByTotalCountAndTotalAmounts=require("./controller/custome
rorder-controller");

const setupAndStartServer = async () => {
  //create express object
  const app = express();

  // parse application/json
  app.use(bodyParser.json());

  // parse application/x-www-form-urlencoded
  app.use(bodyParser.urlencoded({ extended: false }));

  app.get("/customer", getAllCustomers);
  app.get("/order/:id", getOrdersById);

  app.get("/customer-orders",getCustomerOrdersByTotalCountAndTotalAmounts
);

  app.listen(PORT, async () => {
    console.log(`Server started at port ${PORT}`);
  });
};

setupAndStartServer();
```

Question #3: API retrieving list of orders for “customerId” customer.
Ans-
For this also i made a model for it order.js using sequelize
Whose code is this:

```
'use strict';
```

```
const {
  Model
} = require('sequelize');
module.exports = (sequelize, DataTypes) => {
  class Order extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
    //   this.belongsTo(models.Customer,{
    //     foreignKey:'customer_id'
    //   });
    }
  }
  Order.init({
    customer_id: {
        type: DataTypes.INTEGER,
        allowNull: true,
    },
    amount: {
        type: DataTypes.DECIMAL(3, 0),
        allowNull: true,
      },
  }, {
    sequelize,
    modelName: 'order',
    timestamps:false
  });
  return Order;
};
```

And it repository and service layer code is this:

1.repository code

```
```

```
const {order}=require('../models/index');
class OrderRepository{
    async getAllOrders(customerId){
        try {
            const orders=await order.findAll({
                where:{
                    customer_id:customerId
                }
            });
            return orders;
        } catch (error) {
            throw {error};
        }
    }
}
module.exports=OrderRepository;
```

2.service layer code:

```
const OrderRepository=require('../repository/order-repository');
class OrderService{
    constructor(){
        this.orderRepository=new OrderRepository();
    }
    async getOrders(id){
        try {
            const orders=this.orderRepository.getAllOrders(id);
            return orders;
        } catch (error) {
            throw {error};
        }
    }
}
```

```
module.exports=OrderService;
```

And lastly i have a controller to get order by customerid

```
const OrderService=require('../services/order-service');
const orderService=new OrderService();
//get request url=/order
const getOrdersById=async (req,res)=>{
    try {
        const ordersById=await orderService.getOrders(req.params.id);
        return res.status(200).json({
            data:ordersById,
            success:true,
            message:"successfully fetch all the orders of a customer ",
            err:{}
        });
    } catch (error) {
        console.log(error);
        return res.status(500).json({
            data:{},
            success:false,
            message:"failed to get orders",
            err:error
        });
    }
}
module.exports=getOrdersById;
```

And also in first question ans i already have given the index.js code where i have called the route for order using app object using express.

Using this:`app.get("/order/:id", getOrdersById);`

Question #4: API retrieving total number of orders and total amount for each customer, sorted alphabetically by customer country.

Ans for this i used simple raw query in my repository layer and call this method from my controller

Code for that is this :`const { sequelize } = require("../models/index");`

```
class CustomerOrderRepository {
  async getAllCustomerTotalOrdersWithTotalAmount() {
    try {
      const sqlQuery = `SELECT
      c.id AS customer_id,
      c.name AS customer_name,
      c.country AS customer_country,
      COUNT(o.id) AS total_orders,
      SUM(o.amount) AS total_amount
    FROM
      customers c
    LEFT JOIN
      orders o ON c.id = o.customer_id
    GROUP BY
      c.id, c.name, c.country
    ORDER BY
      c.country ASC;`;

      const results = await sequelize.query(sqlQuery, {
        type: sequelize.QueryTypes.SELECT,
      });

      return results;

    } catch (error) {
      throw { error };
    }
  }
}
module.exports = CustomerOrderRepository;
```

And then used this result in my controller `const CustomerOrderService=require('../services/customerorder-serive');`

```
const customerorderService=new CustomerOrderService();
//get request url=/order
const getCustomerOrdersByTotalCountAndTotalAmounts=async (req,res)=>{
    try {
        const results=await customerorderService.getCustomerOrders();
        return res.status(200).json({
            data:results,
            success:true,
            message:"successfully fetch total number of orders and total amount for each customer, sorted alphabetically by customer country. ",
            err:{}
        });
    } catch (error) {
        console.log(error);
        return res.status(500).json({
            data:{},
            success:false,
            message:"failed to fetch the records",
            err:error
        });
    }
}
module.exports=getCustomerOrdersByTotalCountAndTotalAmounts;
```

And finally call the route for it
Which is this

```
app.get("/customer-orders",getCustomerOrdersByTotalCountAndTotalAmounts );
```