**Text Instructions: Module 8**

GitHub Link: <https://github.com/Apollo-Level2-Web-Dev/Level2-Batch4-PH-University-Server/tree/mastering-mongoose>

Installation: <https://blog.logrocket.com/linting-typescript-eslint-prettier>

**9-1 Introduction to validation**

student.model.ts:

import { Schema, model, connect } from 'mongoose';

import { Guardian, LocalGuardian, Student, UserName } from './student/student.interface';

const userNameSchema = new Schema<UserName>({

    firstName: {

        type: String,

        // required: true,

        required: [true, 'First Name is required'],

    },

    middleName: {

        type: String

    },

    lastName: {

        type: String,

        required: [true, 'Last Name is required'],

    }

})

const guardianSchema = new Schema<Guardian>({

    fatherName: {

        type: String,

        required: true,

    },

    fatherOccupation: {

        type: String,

        required: true,

    },

    fatherContactNo: {

        type: String,

        required: true,

    },

    motherName: {

        type: String,

        required: true,

    },

    motherOccupation: {

        type: String,

        required: true,

    },

    motherContactNo: {

        type: String,

        required: true,

    },

})

const localGuardianSchema = new Schema<LocalGuardian>({

    name: {

        type: String,

        required: true,

    },

    occupation: {

        type: String,

        required: true,

    },

    contactNo: {

        type: String,

        required: true,

    },

    address: {

        type: String,

        required: true,

    },

})

const studentSchema = new Schema<Student>({

    id: { type: String , required: true, unique: true},

    name: {

        type: userNameSchema,

        required: true

    },

    gender: {

        type: String,

        enum: {

            values: ['male', 'female', 'other'],

            // message: "The gender field can only be one of the following: 'male', 'female' or 'other' ."

            message: '{VALUE} is not valid'

        },

        required: true

    },

    dateOfBirth: { type: String },

    email: { type: String, required: true },

    contactNo: { type: String, required: true },

    emergencyContactNo: { type: String, required: true },

    bloodGroup: {

        type: String,

        enum: ["A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-"],

    },

    presentAddress: { type: String, required: true },

    permanentAddress: { type: String, required: true },

    guardian: {

        type: guardianSchema,

        required: true

    },

    localGuardian: {

        type: localGuardianSchema,

        required:true

    },

    profileImg: { type: String },

    // isActive: {'active': 'blocked'},

    isActive: {

        type: String,

        enum: ['active', 'blocked'],

        default: 'active',

    },

})

export const StudentModel = model<Student>('Student', studentSchema);