

## //WT CA Test 2

### //Q1 c.

// JS program to validate username , email and age.

```
const prompt = require('prompt-sync')();

function validateUsername(username) {
    const usernameRegex = /^[a-zA-Z0-9]{3,}$/;
    return usernameRegex.test(username); }

function validateEmail(email) {
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return emailRegex.test(email); }

function validatePassword(password) {
    // password should have atleast 6 characters, with one lowercase, one uppercase, one spl
    char and one number.
    const passwordRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-
    z\d@$!%*?&]{6,}$/;
    return passwordRegex.test(password); }

function validateDOB(dob) {
    const today = new Date();
    console.log(today);
    const dobDate = new Date(dob);
    console.log(today.getFullYear());
    const age = today.getFullYear() - dobDate.getFullYear();
    const monthDiff = today.getMonth() - dobDate.getMonth();
    return monthDiff > 0 || (monthDiff === 0 && today.getDate() >= dobDate.getDate()) && age
    >= 18; }

function getUserInput(promptMessage) {
    return prompt(promptMessage); }

const username = getUserInput("Enter username:");
const email = getUserInput("Enter email:");
const password = getUserInput("Enter password:");
const dob = getUserInput("Enter date of birth (YYYY-MM-DD):");
const usernamesValid = validateUsername(username);
const emailsValid = validateEmail(email);
const passwordsValid = validatePassword(password);
const dobsValid = validateDOB(dob);
console.log(`Username is valid: ${usernamesValid}`);
console.log(`Email is valid: ${emailsValid}`);
```

```
console.log(` Password is valid: ${passwordIsValid} `);
console.log(` Date of Birth is valid: ${dobIsValid} `);
const isEligible = usernameIsValid && emailIsValid && passwordIsValid && dobIsValid;
console.log(` Overall Eligibility: ${isEligible ? 'You are eligible for the course!' : 'Sorry, you are
    not eligible for the course.'} `);
//Appreciate Priyadharshini and Vishnupriya of G1
```

**//WT CA 2**

**//Q2. b. ii)**

**//index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head> <body>
  <form action="/" method="post">
    <label for="name"> Name : </label>
    <input type="text" name="name" id="name" required> <br>
    <label for="email"> Email : </label>
    <input type="email" name="email" id="email" required> <br>
    <button type="submit"> Submit </button>
  </form> </body> </html>
```

**//app.js**

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const path = require('path');

const app = express();
```

```

const PORT = 8000;

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

//DB Connection

const db = async() => {
  try {
    await mongoose.connect("mongodb://localhost:27017/testz")
    console.log("db connected");
  } catch (error) {
    console.error(error);    }    }

db();

```

```

//Schema

const UserSchema = new mongoose.Schema({
  name: {
    type:String,
    required: true},
  email: {    type: String,    required: true    }
})

const User = mongoose.model('user',UserSchema)

const createUser = async (req, res) => {
  try {
    const newUser = new User({
      name: req.body.name,
      email: req.body.email,    });
    await newUser.save();
    console.log(newUser);
    res.send("new user created successfully");
  } catch (error) {
    console.error(error);
    res.status(500).send("Error creating user");    }    };

```

```
app.post('/', createUser);
app.use(express.static(__dirname));
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'index.html'));
});

app.listen(PORT, () => {
  console.log(`server is listening to port http://localhost:\${PORT}`); });
```