BICT - Level 1 Semester 1
ICT1153 Web Technologies

Practical 9

1. You are required to create a **Travel Blog Webpage** with a structured layout, styled elements, and responsiveness. The webpage should include a **header, navigation bar, main content section, and footer** with various styling features.

   Create two separate files:

   - index.html – For the webpage structure
   - styles.css – For styling the webpage

   A. Header Section

   - Add a full-width background image with a semi-transparent overlay.
   - Include a title in the header, centered and styled with a custom font.
   - Implement a navigation bar with links to "Home", "Destinations", and "Contact".
   - Make the navigation bar sticky and apply hover effects to the links.

   B. Main Content Section

   - Insert an image of a travel destination, ensuring it has a circular border and a drop shadow.
   - Create a table displaying at least three travel destinations with columns for **"Destination," "Best Time to Visit," and "Activities."**
   - Apply alternate row colors and a hover effect to highlight rows.

   C. Footer Section

   - Design a footer with a gradient background and position it at the bottom of the page.
   - Add links to social media profiles and use icons to represent them.
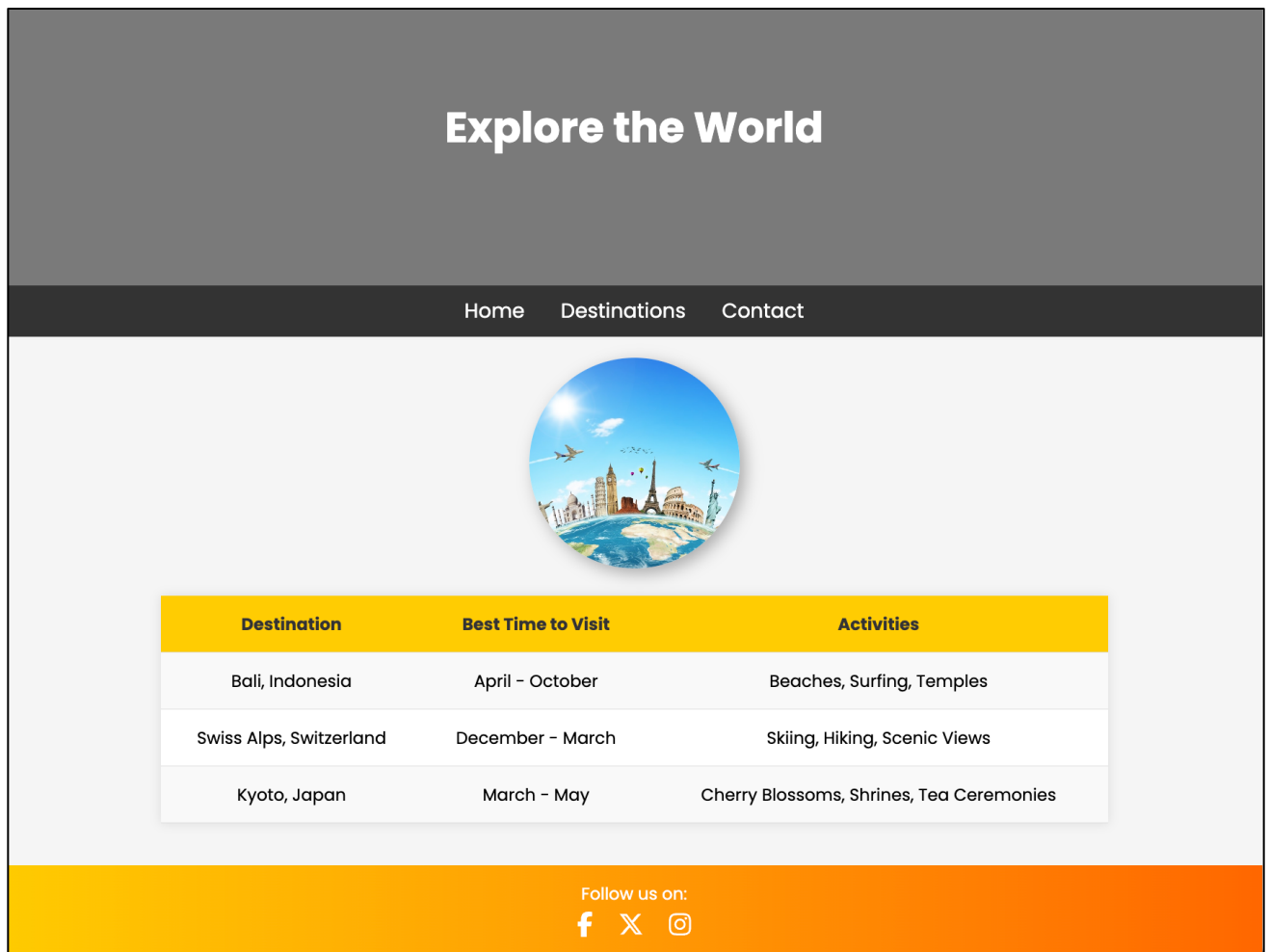   - Implement a hover effect to enlarge the icons when hovered.

Figure 1: Sample Travel Blog Webpage

2. You are tasked with creating a **Personal Portfolio Website with an Advanced Contact Form**. The website should consist of two pages: a **Main Page (index.html)** and a **Contact Page (contact.html)**.

Main Page (index.html with embedded CSS):

1. **Set up the basic HTML structure:**
   - Use the <!DOCTYPE html> declaration.
   - Set the language to English (lang="en").
   - Include the necessary meta tags for character encoding (UTF-8) and responsive design (viewport).
   - Set the title of the page to "Personal Portfolio."

2. **Link external resources:**
   - Add the Poppins font from Google Fonts with weights 300, 400, and 600.
   - Link an external stylesheet named styles.css (even if it's empty).

3. **Add internal CSS for styling:**
   - Set the font family for the entire page to Poppins.
   - Remove default margin and padding from the body and set the background color to #f4f4f4.
   - Center-align all h1, h2, and p elements.
   - Style a button class (.btn) with padding, background color (#007bff), white text, no underline, rounded corners, and a hover effect that changes the background color to #0056b3.

4. **Create the header section:**
   - Add a <header> element with a dark background (#333), white text, and padding.
   - Inside the header, create a <div> with the class overlay to center the content.
   - Include an <h1> element with the developer's name (e.g., "John Doe, Full Stack Software Developer") and a custom color (#ff6f61).
   - Add a <p> element with a short introduction (e.g., "Hi, I'm John Doe, a passionate Full Stack Software Developer") and a custom color (#ffd700).

5. **Create the skills section:**
   - Add a <section> with the class skills and a white background.
   - Include an <h2> element titled "My Skills."
   - Create a table with two columns: "Skill" and "Proficiency Level."
   - Add rows for skills like HTML & CSS, JavaScript, Python, React.js, and SQL & Databases.
   - Use the <progress> element to display proficiency levels (e.g., 90% for HTML & CSS).

6. **Create the contact link section:**
   - Add a <section> with the class contact-link and a white background.
   - Include an <h2> element titled "Get in Touch."
   - Add a <p> element with a message like "Want to collaborate? Feel free to reach out!"
   - Include a button (using the .btn class) that links to a contact.html page with the text "Contact Me."

7. **Ensure the design is clean and responsive:**
   - Use a maximum width of 800px for the header content and center it using margin: 0 auto.
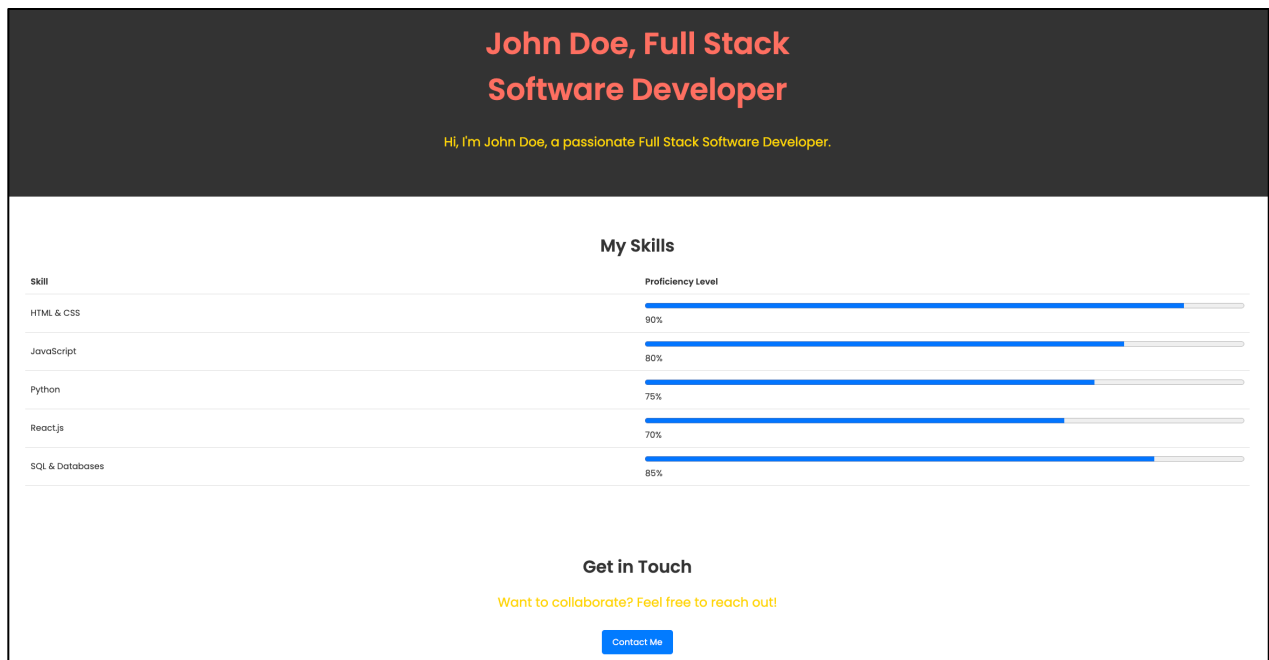   - Set the table width to 80% and center it using margin: 20px auto.

Figure 2: Main Page

Contact Page (contact.html):

1. **Set up the basic HTML structure:**
   o Use the <!DOCTYPE html> declaration.
   o Set the language to English (lang="en").
   o Include the necessary meta tags for character encoding (UTF-8) and responsive design (viewport).
   o Set the title of the page to "Contact Form."

2. **Link external resources:**
   o Add the Poppins font from Google Fonts with weights 300, 400, and 600.
   o Link an external stylesheet named contactPageCSS.css.

3. **Create the form container:**
   o Add a <div> with the class form-container to wrap the form.
   o Include an <h2> element with the text "Contact Us."

4. **Add the form element:**
   o Use the <form> tag with the action attribute set to # and method to POST.

5. **Add basic information fields:**
   o Create a text input for the user's full name (name), with a placeholder and required attribute.

- o Add an email input for the user's email (email), with a placeholder and required attribute.
- o Include a telephone input for the user's phone number (phone), with a placeholder, pattern for 10 digits, and required attribute.

6. **Add date and time preference fields:**
   - o Include a month input for the preferred contact month (contact_month), with a required attribute.
   - o Add a week input for the preferred contact week (contact_week), with a required attribute.
   - o Add a datetime-local input for the preferred date and time (appointment_datetime), with a required attribute.

7. **Add file upload fields:**
   - o Include a file input for uploading a profile picture (profile_picture), accepting only JPG and PNG files, with a required attribute.
   - o Add another file input for uploading a resume (resume), accepting PDF and DOCX files.

8. **Add additional preference fields:**
   - o Include a color input for the user's favorite color (color_preference).
   - o Add a textarea for the user's message (message), with a placeholder, 5 rows, 40 columns, and a required attribute.

9. **Add a multiple-select dropdown:**
   - o Create a <select> element for the contact purpose (purpose), allowing multiple selections and including options like "Job Inquiry," "Collaboration," "Feedback," "Support," and "Other."

10. **Add radio buttons for preferred contact method:**
    - o Include a group of radio buttons for the user to select their preferred contact method (contact_method), with options for "Email," "Phone," and "No Preference" (default).

11. **Add disabled and readonly fields:**
    - o Include a disabled text input for the organization name (disabled_field), with a default value of "Tech Solutions Ltd."
    - o Add a readonly text input for a system-generated reference ID (readonly_info), with a default value of "REF12345678."

12. **Add terms and conditions checkbox:**

o   Include a checkbox for the user to agree to the terms and conditions (terms), with a required attribute and checked by default.

13. **Add a hidden field:**

   o   Include a hidden input field (source) with the value "website_contact_form."

14. **Add a submit button:**

   o   Include a <button> element with the type "submit" to submit the form.

Contact Page Stylesheet (contactPageCSS.css):

1. **Set up the body styles:**
   o   Use the Poppins font for the entire page.
   o   Add a background image (background.jpg) that covers the entire page and is centered.
   o   Set the text color to #333 and remove default margin and padding.

2. **Style the form container:**
   o   Set the background color to a semi-transparent white (rgba(255, 255, 255, 0.8)).
   o   Set the width to 50%, center it using margin: 50px auto, and add padding of 20px.
   o   Add rounded corners (border-radius: 10px) and a subtle shadow (box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2)).

3. **Style the header (**h2**):**
   o   Center-align the text and set the color to #2c3e50.
   o   Use a font size of 2rem and add a bottom margin of 20px.

4. **Style the labels:**
   o   Make labels block-level elements with a font size of 1rem.
   o   Add a bottom margin of 5px and set the color to #2c3e50.

5. **Style text inputs, textareas, and select elements:**
   o   Set the width to 100%, add padding of 10px, and a bottom margin of 15px.
   o   Add a border (1px solid #bdc3c7), rounded corners (border-radius: 5px), and a background color (#ecf0f1).
   o   Use a font size of 1rem and set the text color to #333.
   o   Add a transition effect for the border color (0.3s ease).
   o   On focus, change the border color to #3498db and remove the default outline.

6. **Style the submit button:**
    o   Set the width to 100%, add padding of 12px, and use a background color of #3498db.
    o   Set the text color to white, remove the border, and add rounded corners (border-radius: 5px).
    o   Use a font size of 1rem and add a pointer cursor.
    o   Add a hover effect that changes the background color to #2980b9.

7. **Style radio buttons and checkboxes:**
    o   Create a .radio-group class to display radio buttons in a flex container with space-around alignment.
    o   Add a bottom margin of 15px to the radio group.
    o   Style the labels of radio buttons to be inline with a right margin of 20px.

8. **Style the form layout:**
    o   Use a flexbox layout for the form container to align items vertically (flex-direction: column).
    o   Ensure all form fields stretch to fill the container width.

9. **Style the multiple-select dropdown:**
    o   Set the height to auto and add a maximum height of 200px.

10. **Style disabled and readonly fields:**
    o   Set the background color of disabled fields to #bdc3c7.
    o   Set the background color of readonly fields to #ecf0f1.

11. **Add spacing and alignment:**
    o   Add a top margin of 10px to labels, inputs, textareas, selects, and buttons.

12. **Make the form responsive:**
    o   On screens smaller than 768px, set the form container width to 90%.

13. **Add custom colors for specific labels:**
    o   Use attribute selectors to assign unique colors to labels for fields like name, email, phone, contact_month, contact_week, appointment_datetime, profile_picture, resume, color_preference, message, purpose, and terms.

Contact Us

Full Name

Enter your full name

Email

Enter your email

Phone Number

Enter your phone number

Preferred Month for Contact

---------- ----

Preferred Week for Contact

Week --, ----

Preferred Date and Time

yyyy-mm-dd, --:--

Upload Profile Picture (JPG/PNG)

Choose file | No file chosen

Upload Resume (PDF/DOCX)

Choose file | No file chosen

Favorite Color (For Design Preferences)

Message

Write your message

Contact Purpose (Select Multiple)

Job Inquiry
Collaboration
Feedback
Support
Other

Preferred Contact Method

○ Email   ○ Phone   ◉ No Preference

Organization (Not Editable)

Tech Solutions Ltd.

System Generated Reference ID

REF12345678

☑ I agree to the terms and conditions

Submit

Figure 3: Contact Page

8

NOTE:

Poppins font

The Poppins font is a popular sans-serif typeface available on Google Fonts. You can easily include it in your HTML page to use for text styling. Here's how you can incorporate the Poppins font into your web project:

Step 1: Include the Poppins Font Link

You can include the Poppins font from Google Fonts by adding the following `<link>` tag in the `<head>` section of your HTML document:

```
<head>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=s
wap" rel="stylesheet">
</head>
```

- `family=Poppins:wght@300;400;600` specifies that you want to load three different font weights (300, 400, and 600). You can adjust these weights according to your needs.
- `display=swap` ensures that the font is swapped as soon as it loads, preventing the layout from jumping.

Step 2: Apply the Poppins Font Using CSS

Once you have included the font in your HTML, you can apply it to your elements using CSS:

```
body {
    font-family: 'Poppins', sans-serif;
}
```

This will apply the **Poppins** font to the entire body text on your page. You can also apply it to specific elements:

```
h1, h2, h3 {
    font-family: 'Poppins', sans-serif;
}
```

<progress> element

The `<progress>` element in HTML represents the completion of a task or process, usually in the form of a progress bar. It is commonly used for indicating the status of a task such as file uploads, downloads, or background processes.

Syntax:

```
<progress value="50" max="100"></progress>
```

9

- `value`: Specifies the current value of the progress bar (how much of the task is completed).
- `max`: Specifies the maximum value of the progress bar (the total amount the task will require).
- The value is usually a percentage (0 to 100), but it can also be any number based on your needs.