Lesson Plan: Grade 9 ICT - From Containers to a Responsive Page

Student: Grade 9 student **Topic:** Bootstrap 5 Fundamentals (Containers, Grid, Images, Tables) **Tools:** Visual Studio Code, Web Browser **Files:** `democontainerTAN.html` , `sw 1.pdf` **Time:** ~1 Hour

Session Objective

By the end of this session, the student will:

1. Understand the difference between Sublime Text and VS Code for web development.
2. Explain the purpose of Bootstrap's `container` , `container-fluid` , and responsive containers.
3. Understand and use Bootstrap's **Grid System** (`.row` , `.col`) to build layouts.
4. Apply Bootstrap classes to style images (`.img-fluid` , `.rounded`) and tables (`.table`).
5. Feel confident starting their "Favorite Artists" seatwork (`sw 1.pdf`).

Lesson Flow

Part 1: Why VS Code? (10 mins)

Start by opening their `democontainerTAN.html` file in VS Code.

1. **"First, let's see why you asked about VS Code. It's a great tool."**
2. **Point out key features:**
   - **Syntax Highlighting:** "Notice how all the HTML tags, attributes, and text are different colors? VS Code does this automatically to make code easier to read than in a plain editor."
   - **File Explorer:** "On the side, we have a list of all our files. This is great when your project gets bigger than one file."
   - **IntelliSense (Code Completion):** "This is the best part. When you start typing in the `class=""` attribute, VS Code will suggest Bootstrap classes for you. Sublime can't do that." (Be ready to demo this in Part 3).
3. **Introduce the "Live Server" Extension:**
   - Go to the "Extensions" tab in VS Code (the four squares).
   - Search for and install "Live Server" (by Ritwick Dey).
   - **Explain:** "In Sublime, you have to save your file, then go to your browser and hit 'refresh' every time, right? With Live Server, we just right-click our HTML file and click 'Open with Live Server'. It opens in your browser, and *automatically* refreshes every time we save. It's a huge time-saver."
   - **Action:** "Go Live" with `democontainerTAN.html` .

Part 2: What We Know - The `container` Demo (15 mins)

With `democontainerTAN.html` open in the browser (via Live Server), review what it's teaching.

1. **The `<head>` Section:**
   - Point to the `<link ... bootstrap.min.css>` : "This is the most important line. It's what 'activates' Bootstrap. If we comment this out, all our styling is gone." (Demo this by adding `<!--` and `-->` around it and saving. Live Server will instantly show the broken page. Then undo it).
   - Point to `<meta name="viewport"` : "This line is critical for mobile. It tells the phone's browser, 'Don't zoom out. Show the page at its real size.' All responsive pages need this."
2. **The `container` Classes:**
   - Keep the browser window and the code side-by-side.
   - `.container-fluid` **(Red):** "See this? It's *always* 100% width. It goes from one edge of the screen to the other, no matter how wide you make the window."
   - `.container` **(Basic - not in the demo but good to add):** "A basic `.container` will be 100% wide until it hits a certain *breakpoint*, then it will stay at a fixed width and center itself."
   - `.container-sm` , `-md` , `-lg` , `-xl` **(Orange, Yellow, etc.):** "These are just like the basic `.container` , but they stay 100% width for *longer*. The `.container-lg` (green) stays 100% width on small and medium screens, but once the screen is 'large' (992px), it stops growing. This gives you control over your layout."
   - **Key Takeaway:** "The `container` is the main box for all your page content. You have to choose: do you want it to be full-width (`.container-fluid`) or 'boxed' and centered (`.container`)?"

Part 3: What We Need - Building the "Favorite Artists" Page (30 mins)

This is where you connect the dots. The seatwork sample shows 5 items in a row.

1. **"How do we get 5 items in a row?"**
   - "The seatwork sample shows 5 artists side-by-side. The `container` classes we just saw don't do that. For that, we need Bootstrap's **Grid System**."
   - "The grid is simple. It has two parts: **Rows** (`.row`) and **Columns** (`.col`). A row is a horizontal container, and the columns go inside it. You get 12 columns per row."
2. **Live-Code a New File (`artists.html`):**
   - Create a new file. Use VS Code's Emmet shortcut: type `!` and hit `Enter` to get a blank HTML template.
   - Add the Bootstrap CSS link and `meta viewport` tag (copy from the demo).
   - **Explain as you type:**

```
<body>
  <!-- Let's use a 'boxed' container for this page -->
  <div class="container">

    <!-- A heading for the page. 'text-center' and 'my-4' are Bootstrap classes
    <h1 class="text-center my-4">My Favorite Artists</h1>

    <!-- 1. We start with a .row to hold our columns -->
    <div class="row">

      <!-- 2. We add .col for each artist.
           If we just use '.col', Bootstrap is smart and
           will make all 5 fit and be equal-width.
      -->
      <div class="col">
        <!-- Artist 1 content will go here -->
        <p>Artist 1</p>
      </div>
      <div class="col">
        <p>Artist 2</p>
      </div>
      <div class="col">
        <p>Artist 3</p>
      </div>
      <div class="col">
        <p>Artist 4</p>
      </div>
      <div class="col">
        <p>Artist 5</p>
      </div>

    </div> <!-- end of .row -->
  </div> <!-- end of .container -->

  <!-- Don't forget the Bootstrap JS file at the bottom! -->
  <script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.
</body>
```

- **Show this in Live Server.** They'll see 5 items in a row.
- **Crucial Moment:** Resize the browser window to be small (like a phone). "Watch this! On a small screen, the columns *automatically stack* on top of each other. That's responsiveness! You get it for free with the grid."

3. **Add Seatwork Elements (Images & Tables):**
    - "Okay, let's build out one artist, like the seatwork sample. It has an image and a table."
    - **Bootstrap Images:**
        - "Let's add an image to Artist 1. I'll grab a URL from the web."
        - Add a plain `<img>` tag. It will probably be huge and break the layout.
        - "See? It's not responsive. But if we add the Bootstrap class `img-fluid` , it will automatically fit its container (the `.col`)."
        - "The sample has round images. We can add the `rounded-circle` class too."

- Bootstrap Tables:
  - "The sample has a 'Greatest Hit' table. Let's add one."
  - Add a standard HTML `<table>`. It will look plain.
  - "Now, just add the `.table` class to the `<table>` tag."
  - "Instantly, it looks better. You can also add `.table-striped` (for zebra-striping) or `.table-dark`."
- Update the code for Artist 1:

```
<div class="col text-center">
  <!-- The 'text-center' class on the .col will center everything in it -->

  <!-- Add 'img-fluid' and 'rounded-circle' -->
  <img src="https...image-url-here.jpg" alt="Artist Name"
       class="img-fluid rounded-circle" style="width: 150px;">

  <h3 class="mt-2">Justin Bieber</h3> <!-- 'mt-2' means 'margin-top' -->

  <!-- Add 'table' and 'table-striped' -->
  <table class="table table-striped mt-3">
    <thead>
      <tr>
        <th>Greatest Hit</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Baby</td>
      </tr>
    </tbody>
  </table>
</div>
```

(Self-note: While the demo is simple, you could also mention that the "Card" component is the "pro" way to do this, as it bundles images and text.)

Part 4: Your Turn & Wrap-up (5-10 mins)

1. **Review:**
   - "So, to build your page, what are the main parts you need?"
   - Answer: `.container` (to hold everything), `.row` (to hold the columns), and `.col` (one for each artist).
2. **Key Classes:**
   - For responsive images: `img-fluid`
   - For round images: `rounded-circle`
   - For a basic table: `table`

3. **Student's Task:** "Now you try. Copy the code we just wrote for Artist 1 into the `div` for Artist 2. Then, all you have to do is change the image URL and the text in the table."
4. **Final Check:** "Does this make sense? You can see how these simple classes let you build a pretty good-looking, responsive page really fast. That's the power of Bootstrap."