4. Develop a Progressive Web Application (PWA) using HTML, CSS andJavaScript.

Progressive web apps are a way to bring that native app feeling to a traditional web app. WithPWAs we can enhance our website with mobile app features which increase usability and offera great user experience. It gives you the ability,

- To install it on a mobile home screen
- To access it when offline
- To access the camera
- To get push notifications
- To do background synchronization

PWA has five technical components,

1. **Web App manifest**

   The web app manifest is the first component of the PWA. It is a simple JSON file thatcontrols a user's application. Usually, it is named "manifest.json". It is the most important component for the presence of PWA. When you first connect PWA to a network, a mobile browser reads the "manifest.json" file and stores it locally in cache memory.

2. **Application shell**

   It is specialized to split the static and dynamic content of the application. The minimal HTML, CSS, JavaScript and any other dynamic and static resources offer thestructure for your web page. It reduces the actual content that is unique to the webpage.This component ensures a very critical approach to the development of progressive webapps.

3. **Service worker**

   A service worker is a web worker. It is a JavaScript file that runs aside from the mobile browser. In other words, it is another technical component that promotes the functionality of PWA. The service worker retrieves the resources from the cache memory and delivers the messages.

4. **Webpack**

   It is used to design the PWA front-end. It allows the PWA-developers to gather all JavaScript resources and data in one location.

5. **Transport Layer Security (TLS)**

   This component is a standard for all robust and secure data exchange between any twoapplications. The integrity of the data requires the website's service through the HTTPSand an SSL certificate installed on the server.

Create a new project using the following structure. Create a new folder for your project and setup the following files inside it:

- **pgrm4.html**: The main HTML file for your PWA.
- **pgrm4.css**: The CSS file to style your application.
- **app.js**: The JavaScript file for handling interactivity and service workers.
- **service-worker.js**: The service worker file.

prgm4.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      body {
        background-color: beige;
      }

      form {
        display: flex;
        align-items: center;
        justify-content: center;
      }

      h2 {
        text-align: center;
      }

      #hi {
        height: 20px;
        font-size: 20px;
        padding-left: 9px;
      }

      .li {
        padding-top: 10px;
        font-size: 34px;
        margin: 100px;
      }

      ul {
        font-size: 34px;
      }
    </style>
  </head>
<body>
    <br /><br />
    <h2>To Do App</h2>
    <br />
    <form class="formm" onsubmit="return addtask()">
      <input type="text" id="hi" placeholder="Enter the task" />
      <input type="submit" value="AddTask" />
    </form>
    <ul id="listed"></ul>
```

```
<script>
      function addtask() {
        let task = document.getElementById("hi").value;
        let list = document.getElementById("listed");

        let list_item = document.createElement("li");
        list_item.textContent = task;

        list.appendChild(list_item);

        list_item.addEventListener("click", function () {
          list_item.style.textDecoration = "line-through";
        });

        document.getElementById("hi").value = "";

        return false;
      }
    </script>
  </body>
</html>
```