WEB TECHNOLOGY LABORATORY (22CSL35))

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

Progressive web apps are a way to bring that native app feeling to a traditional web app. With PWAs we can enhance our website with mobile app features which increase usability and offer a 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 that controls 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 the structure 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 web apps.

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 two applications. The integrity of the data requires the website's service through the HTTPS and an SSL certificate installed on the server.

Create a new project using the following structure. Create a new folder for your project and set up 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>
//<link rel="manifest" href="/Users/amithpradhaan/Documents/ABILITY ENHANCEMENT (WEB TECHNOLOGY)/Pgrm4/manifest.json">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ToDo App</title>
  <link rel="stylesheet" href="pgrm4.css">
</head>

<body>
  <header>
    <h1>ToDo App</h1>
  </header>

  <main>
    <form id="todo-form">
      <input type="text" id="todo-input" placeholder="Enter a task">
      <button type="submit">Add</button>
    </form>
    <ul id="todo-list"></ul>
  </main>

  <script src="app.js"></script>
</body>

</html>
```

prgm4.css

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

header {
    background-color: #f2f2f2; padding:
    20px;
    text-align: center;
}

h1 { margin: 0;
}

form { display:
    flex;
    margin-bottom: 20px;
}

input[type="text"] { flex:
    1;
    padding: 10px;
}

button {
    padding: 10px 20px;
    background-color: #4CAF50; color:
    #fff;
    border: none;
    cursor: pointer;
}
```

```
ul {
  list-style-type: none;
  padding: 0;
}

li {
  display: flex;
  align-items: center;
  padding: 10px;
  border-bottom: 1px solid #ccc;
}

li:last-child {
  border-bottom: none;
}

.completed {
  text-decoration: line-through;
}
```

app.js

```
// Register Service Worker
if ('serviceWorker' in navigator) {
    navigator.serviceWorker
      .register('service-worker.js')
      .then(function() {
        console.log('Service Worker Registered');
      });
}

  // DOM elements
  const todoForm = document.getElementById('todo-form'); const todoInput =
  document.getElementById('todo-input');
```

```
const todoList = document.getElementById('todo-list');

// Store tasks in an array let
tasks = [];

// Render tasks in the list function
renderTasks() { todoList.innerHTML
= '';

  tasks.forEach(function(task) {
    const li = document.createElement('li');
    li.textContent = task.title;

    if (task.completed) {
      li.classList.add('completed');
    }

    li.addEventListener('click', function() {
      task.completed = !task.completed; renderTasks();
    });

    todoList.appendChild(li);
  });
}

// Handle form submission todoForm.addEventListener('submit',
function(event) { event.preventDefault();

  const taskTitle = todoInput.value.trim();

  if (taskTitle !== '') { const
    task = {
```

```
        title: taskTitle,
        completed: false
      };

      tasks.push(task);

      renderTasks();

      todoInput.value = '';
    }
  });
```

service-worker.js

```
// service-worker.js
const CACHE_NAME = 'my-pwa-cache-v1'; const
urlsToCache = [
  '/',
  'pgrm4.html',
  'prgm4.css',
  'app.js'
];

self.addEventListener('install', (event) => {
  event.waitUntil( caches.open(CACHE_NAME)
      .then((cache) => cache.addAll(urlsToCache))
  );
});

self.addEventListener('fetch', (event) => {
  event.respondWith( caches.match(event.request)
```

```
      .then((response) => response || fetch(event.request))
  );
});
```