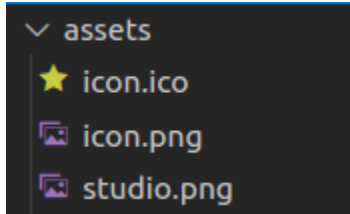


7. Create a login page to authenticate a user using PWA with Manifest file

Step 1: Place all the icons in the assets directory

Note: Icons must be square and greater than or equal to 144x144px



Step 2: Create the manifest file

manifest.json

```
{
  "name": "User Authentication",
  "short_name": "Login",
  "display": "standalone",
  "start_url": "/",
  "id": "/",
  "Orientation": "portrait_primary",
  "description": "Login using this application",
  "icons": [
    {
      "src": "assets/icon.png",
      "type": "image/png",
      "sizes": "512x512"
    },
    {
      "src": "assets/studio.png",
      "type": "image/png",
      "sizes": "128x128"
    }
  ],
  "background_color": "#0000FF",
  "theme_color": "#0000CC"
}
```

Step 3: Create the authentication page in index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
    // linking the manifest file
    <link rel="manifest" href="manifest.json" />
    <link rel="stylesheet" href="index.css" />
    <link rel="icon" type="image/x-icon" href="assets/icon.ico" />
  </head>
  <body>
    <center>
      <h2>Login to the Application</h2>
      <table>
        <form id="form">
          <tr>
            <th>Username:</th>
            <td>
              <input id="username" type="text" required />
            </td>
          </tr>
          <tr>
            <th>Password:</th>
            <td>
              <input id="password" type="password" required />
            </td>
          </tr>
          <tr>
            <td colspan=2 align="center">
              <input type="submit" id="submit" value="Login" />
            </td>
          </tr>
        </form>
      </table>
    </center>
    <script>
      // storing username and password in local storage
      localStorage.setItem("username", "qwerty");
      localStorage.setItem("password", "123456");
    </script>
  </body>
</html>
```

```

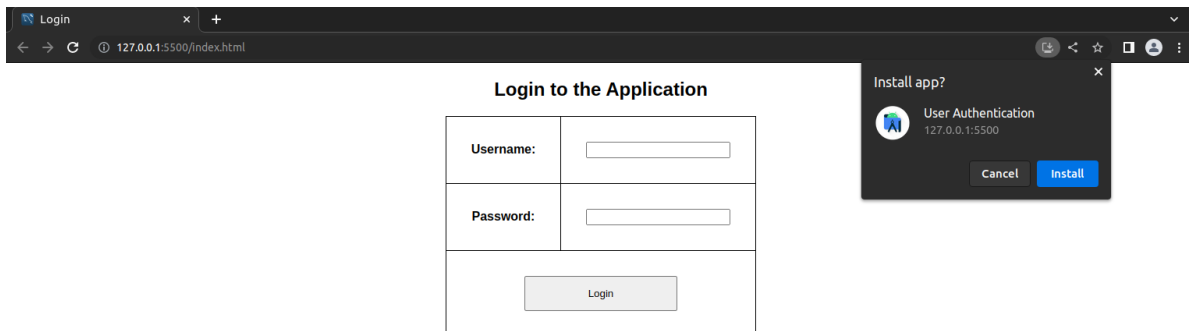
const login = (e) => {
    e.preventDefault();
    // getting the username and password from form inputs
    var un = document.getElementById("username").value;
    var ps = document.getElementById("password").value;
    // getting the username and password from local storage
    var username = localStorage.getItem("username");
    var password = localStorage.getItem("password");

    if(username == un && password == ps) {
        alert("Login successful!");
    } else {
        alert("Login failed!");
    }
}

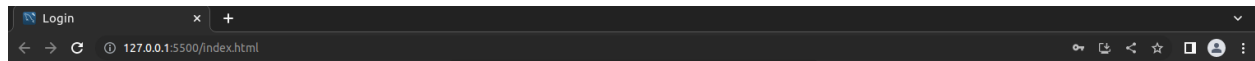
// running the 'login' function on form submit event
document.getElementById("form").addEventListener("submit",
login);
</script>
</body>
</html>

```

Output-



The screenshot shows a web browser window with the title "Login". The address bar displays "127.0.0.1:5500/index.html". The main content area features a login form titled "Login to the Application". The form has two input fields: "Username:" and "Password:", each followed by a text input box. Below these fields is a "Login" button. To the right of the form, there is a small dialog box titled "Install app?". The dialog box contains a circular icon with a blue and green design, the text "User Authentication", and the version "127.0.0.1:5500". At the bottom of the dialog box are two buttons: "Cancel" and "Install".



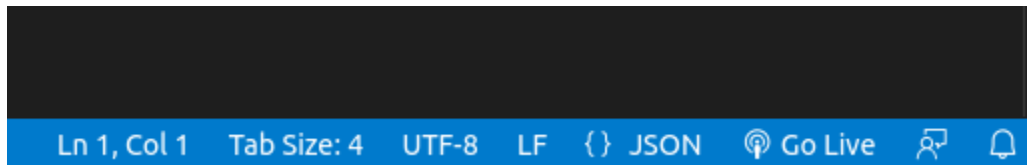
127.0.0.1:5500 says Login Failed!	
<input type="button" value="OK"/>	
Password:	<input type="password" value="****"/>
<input type="button" value="Login"/>	



127.0.0.1:5500 says Login successful!	
<input type="button" value="OK"/>	
Password:	<input type="password" value="*****"/>
<input type="button" value="Login"/>	

Running the program-

Run the live server in VS Code by pressing the Go Live button



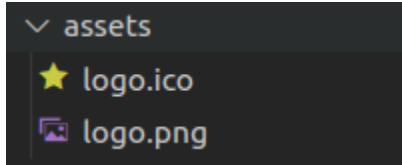
To view the output on the Android Studio Emulator-

1. Run the AVD
2. Open the AVD settings
3. Search for proxy settings
4. Get the ip address of the system using `ifconfig` and set it as proxy with port 5500
5. Open chrome in the emulator and enter the ip address of the system as url

8. Build a simple web page using PWA by adding a Service Worker

Step 1: Place all the icons in the assets directory

Note: Icons must be square and greater than or equal to 144x144px



Step 2: Create the manifest file

manifest.json

```
{
  "name": "Music Academy",
  "short_name": "Registration",
  "display": "standalone",
  "start_url": "/",
  "id": "/",
  "Orientation": "portrait_primary",
  "description": "Register to the Music Academy",
  "icons": [
    {
      "src": "assets/logo.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "background_color": "#00AA00",
  "theme_color": "#00CC00"
}
```

Step 3: Create the service worker file

sw.js

```
self.addEventListener("install", (e) => {
  console.log("installed");
})

self.addEventListener("activate", (e) => {
  console.log("activated");
})
```

```
self.addEventListener("fetch", (e) => {
  console.log("fetched");
}))
```

Step 3: Create any simple page in index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Register</title>
    // linking the manifest file
    <link rel="manifest" href="manifest.json" />
    <link rel="stylesheet" href="index.css" />
    <link rel="icon" type="image/x-icon" href="assets/logo.ico" />
  </head>
  <body>
    <center>
      <h2>Register to the the Music Academy</h2>
      <table>
        <form id="form">
          <tr>
            <th>USN:</th>
            <td>
              <input id="usn" type="text"
pattern="1RV[0-9]{2}[A-Z]{2}[0-9]{3}" required />
            </td>
          </tr>
          <tr>
            <th>Name:</th>
            <td>
              <input id="name" type="text" required />
            </td>
          </tr>
          <tr>
            <th>Place:</th>
            <td>
              <input id="place" type="text" required />
            </td>
          </tr>
        </form>
      </table>
    </center>
  </body>
</html>
```

```

        <tr>
            <th>No of days:</th>
            <td>
                <input id="nod" type="number" min="4" max="14"
required />
            </td>
        </tr>
        <tr>
            <td colspan=2 align="center">
                <input type="submit" id="submit"
value="REGISTER" />
            </td>
        </tr>
    </form>
</table>
</center>
<script>
    // checking if service workers are supported by the browser
    if('serviceWorker' in navigator) {
        // registering the service worker
        navigator.serviceWorker.register("sw.js")
        .then((e) => {
            console.log("registered");
        });
    } else {
        console.log("browser does not support service worker");
    }

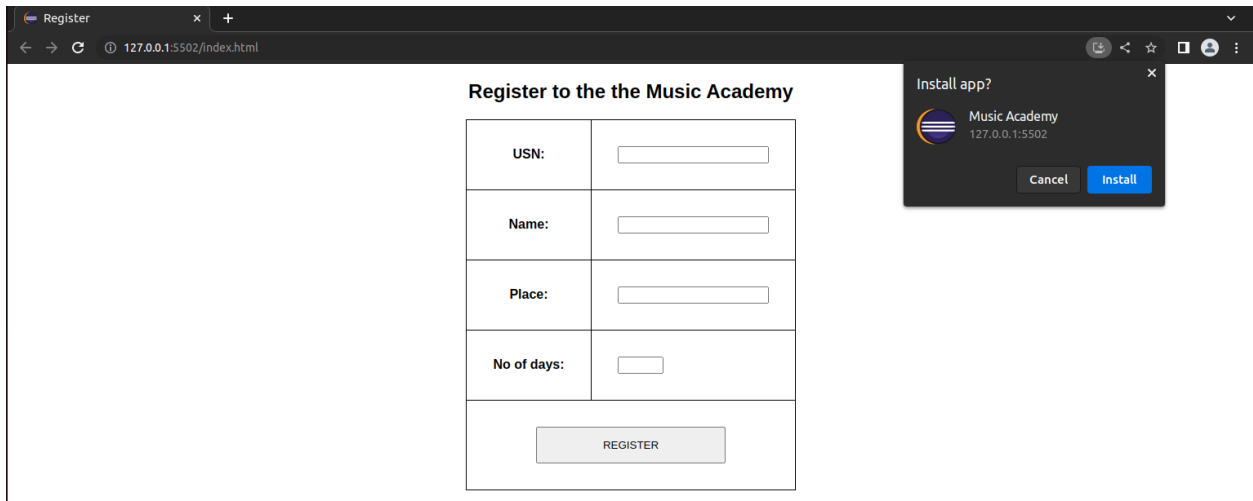
    const register = (e) => {
        e.preventDefault();

        alert("Registered!");
    }

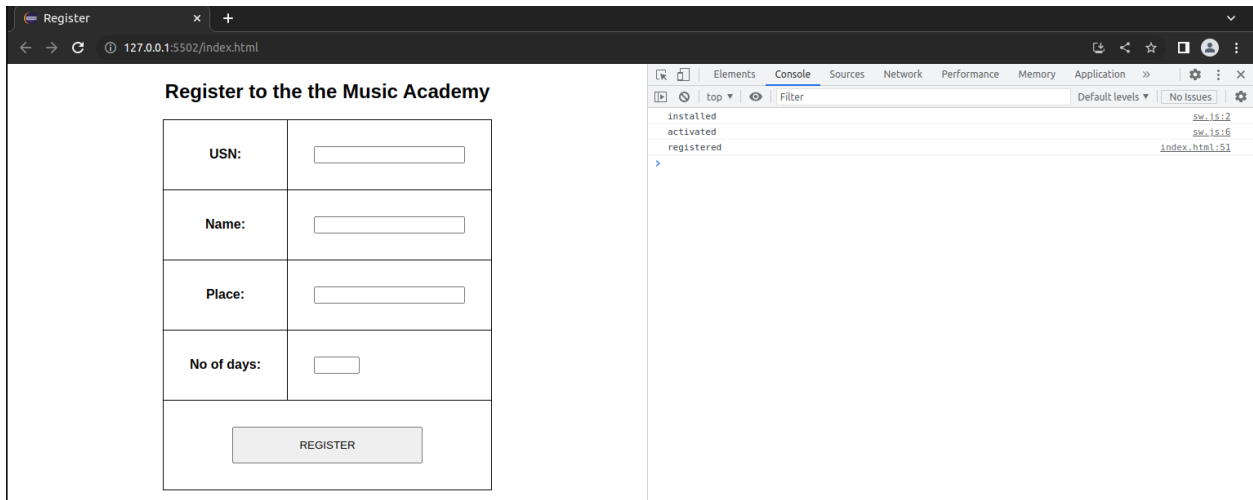
    document.getElementById("form").addEventListener("submit",
register);
</script>
</body>
</html>

```

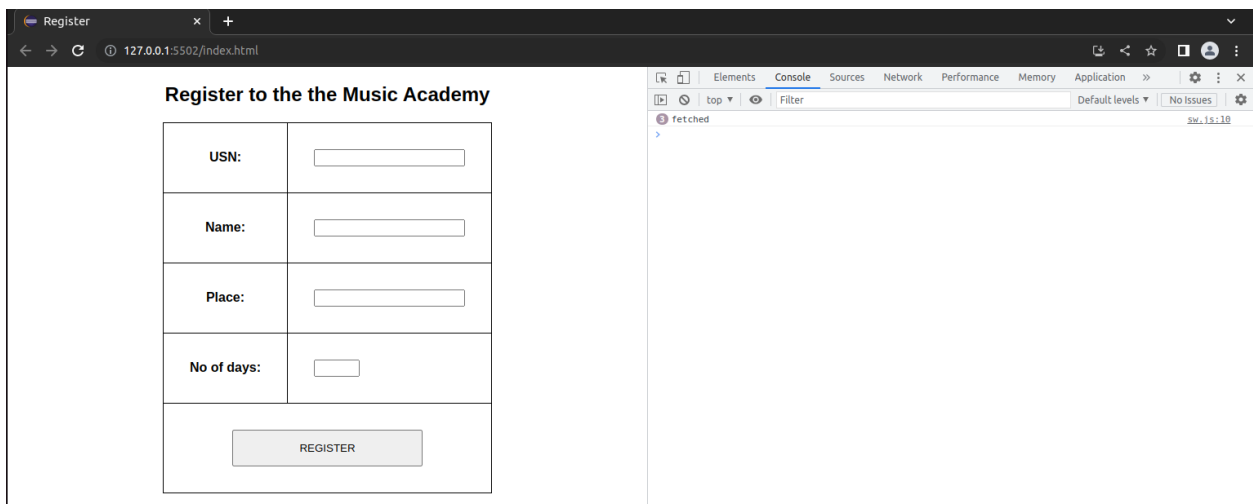
Output-



On loading for the first time- registered, installed, activated must appear



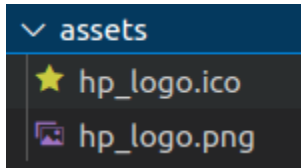
On reload- fetched must appear



9. Devise a PWA to fetch the Github Details for a particular person using the Fetch API

Step 1: Place all the icons in the assets directory

Note: Icons must be square and greater than or equal to 144x144px



Step 2: Create the manifest file

manifest.json

```
{
  "name": "Fetch Github Details",
  "short_name": "API Fetch",
  "display": "standalone",
  "start_url": "/",
  "id": "/",
  "Orientation": "portrait_primary",
  "description": "Get user details from Github using it's API",
  "icons": [
    {
      "src": "assets/hp_logo.png",
      "type": "image/png",
      "sizes": "256x256"
    }
  ],
  "background_color": "#AA0000",
  "theme_color": "#CC0000"
}
```

Step 3: Create the service worker file

sw.js

```
self.addEventListener("install", (e) => {
  console.log("installed");
})

self.addEventListener("activate", (e) => {
  console.log("activated");
})
```

```

self.addEventListener("fetch", (e) => {
  console.log("fetched");
  // fetching user data from the github api
  fetch("https://api.github.com/users/your_github_username")
    .then((e) => {
      return e.json();
    })
    .then((json) => {
      console.log(json);
    })
    .catch((e) => {
      console.log(e);
    });
});
})

```

Step 4: Create a html page to register the service worker

index.html

```

<!DOCTYPE html>
<html>
  <head>
    <title>Register</title>
    // linking the manifest file
    <link rel="manifest" href="manifest.json" />
    <link rel="stylesheet" href="index.css" />
    <link rel="icon" type="image/x-icon" href="assets/hp_logo.ico" />
  </head>
  <body>
    <center>
      <h2>Fetch From Github API</h2>
    </center>
    <script>
      // checking if service workers are supported by the browser
      if('serviceWorker' in navigator) {
        // registering the service worker
        navigator.serviceWorker.register("sw.js")
          .then((e) => {
            console.log("registered");
          });
      } else {

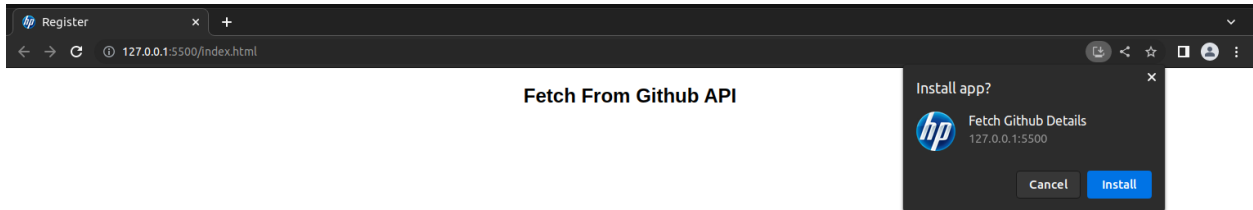
```

```

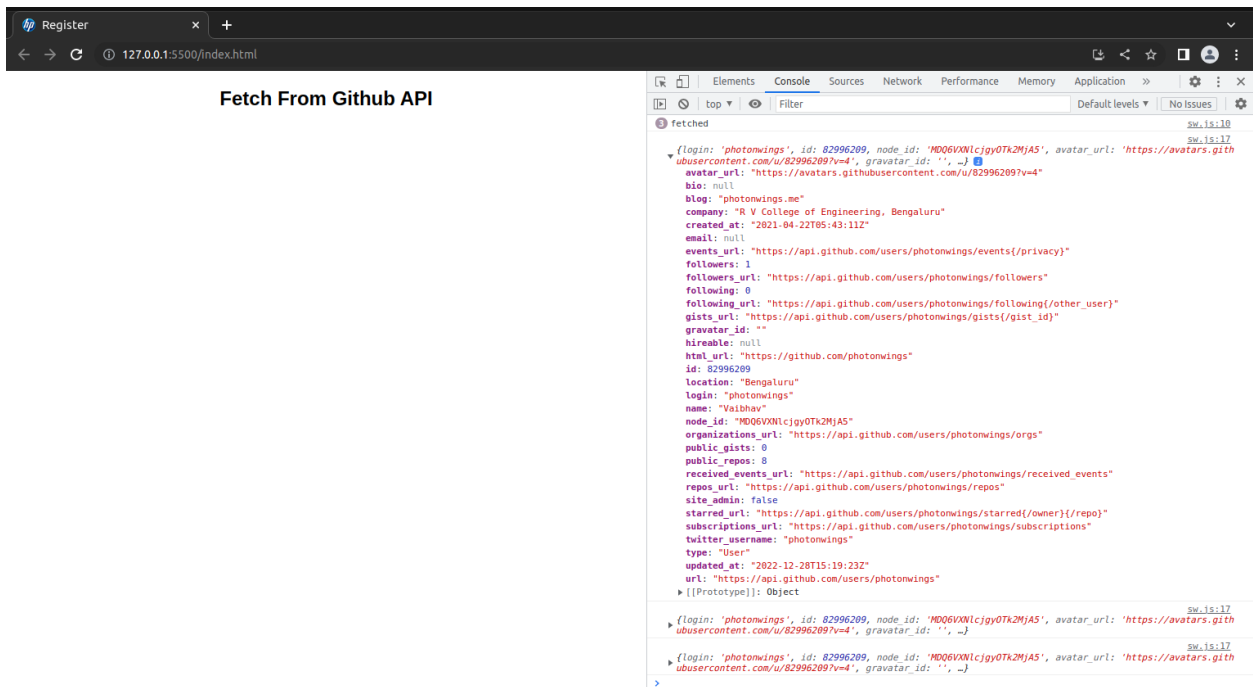
        console.log("browser does not support service worker");
    }
</script>
</body>
</html>

```

Output- (make sure to unregister the service worker from the previous programs in the applications tab of the inspect menu)



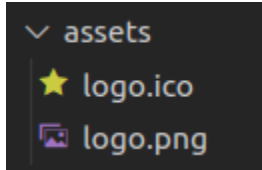
On reload-



10. Build an application to do a stock display using PWA using a raw JSON file

Step 1: Place all the icons in the assets directory

Note: Icons must be square and greater than or equal to 144x144px



Step 2: Create the manifest file

manifest.json

```
{
  "name": "Stock Display",
  "short_name": "Stock-Display",
  "display": "standalone",
  "start_url": "/",
  "id": "/",
  "description": "Display stock from json file",
  "icons": [
    {
      "src": "assets/logo.png",
      "type": "image/png",
      "sizes": "512x512",
      "purpose": "maskable"
    },
    {
      "src": "assets/logo.png",
      "type": "image/png",
      "sizes": "512x512",
      "purpose": "any"
    }
  ],
  "background_color": "#00AA00",
  "theme_color": "#00CC00"
}
```

Step 3: Create the service worker file

sw.js

```
self.addEventListener("install", (e) => {
  console.log("installed");

  // creating a cache storage and adding all the files required for the
  web page to run even when it's offline
  caches.open("stock").then((cache) => {
    cache.add("/");
    cache.add("./assets/logo.ico");
    cache.add("./assets/logo.png");
    cache.add("./data.json");
    cache.add("./index.html");
    cache.add("./manifest.json");
    cache.add("./sw.js");
  })
  .catch((err) => {
    console.log(err);
  });
});

self.addEventListener("activate", (e) => {
  console.log("activated");
});

self.addEventListener("fetch", (e) => {
  console.log("fetched");
  // intercepting every fetch request and responding from the cache when
  it's offline
  e.respondWith(
    // checking if requested data is present in the cache
    caches.match(e.request)
    .then((res) => {
      // respond from cache when fetch function doesn't return any
      response(i.e, the system is offline)
      return res || fetch(e.request);
    })
    .catch((err) => {
      console.log(err);
    })
  );
});
```

```
);  
})
```

Step 4: Create a json file and fill some stock market data

Sample data - <https://gist.github.com/tanveery/4ac939d2ad27954da4c8db13e10ef7bd>

data.json

```
[  
  {  
    "company": "3M",  
    "description": "3M, based in Minnesota, may be best known for its  
Scotch tape and Post-It Notes, but it also produces sand paper, adhesives,  
medical products, computer screen filters, food safety items, stationery  
products and many products used in automotive, marine, and aircraft  
industries.",  
    "initial_price": 44.28,  
    "price_2002": 56.27,  
    "price_2007": 95.85,  
    "symbol": "MMM"  
  },  
  {  
    "company": "Amazon.com",  
    "description": "Amazon.com, Inc. is an online retailer in North  
America and internationally. The company serves consumers through its  
retail Web sites and focuses on selection, price, and convenience. It also  
offers programs that enable sellers to sell their products on its Web  
sites, and their own branded Web sites. In addition, the company serves  
developer customers through Amazon Web Services, which provides access to  
technology infrastructure that developers can use to enable virtually  
various type of business. Further, it manufactures and sells the Kindle  
e-reader. Founded in 1994 and headquartered in Seattle, Washington.",  
    "initial_price": 89.38,  
    "price_2002": 17.01,  
    "price_2007": 93.43,  
    "symbol": "AMZN"  
  },  
  .  
  .  
  .  
]
```

```
.  
.   
.   
.   
]
```

Step 5: Create a html page to register the service worker and display the stock market data
index.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Stock Display</title>  
    <link rel="manifest" href="manifest.json">  
    <style>  
      table{  
        border-collapse: collapse;  
      }  
      td, th {  
        padding: 1em;  
        border: 1px solid black;  
      }  
    </style>  
  </head>  
  <body>  
    <center>  
      <h2>Stock Market Data Display</h2>  
      <div id="stock"></div>  
    </center>  
    <script>  
      if('serviceWorker' in navigator) {  
        navigator.serviceWorker.register("sw.js")  
          .then((e) => {  
            if(e.active === null){  
              console.log("registered");  
            }  
          });  
      } else {  
        console.log("browser does not support service worker");  
      }  
    </script>  
  </body>  
</html>
```

```

        // data from
https://gist.github.com/tanveery/4ac939d2ad27954da4c8db13e10ef7bd
        fetch("data.json")
        .then((res) => {
            return res.json();
        })
        .then((stock) => {
            // creating the table for displaying the stock data
            var table = "<table>";
            table +=
"<tr><th>Company</th><th>Description</th><th>Initial Price</th><th>Price
in 2007</th><th>Symbol</th></tr>";

            for(item of stock) {
                table +=
"<tr><td>" + item.company + "</td><td>" + item.description + "</td><td>" + item.init
ial_price + "</td><td>" + item.price_2007 + "</td><td>" + item.symbol + "</td></tr>"
;

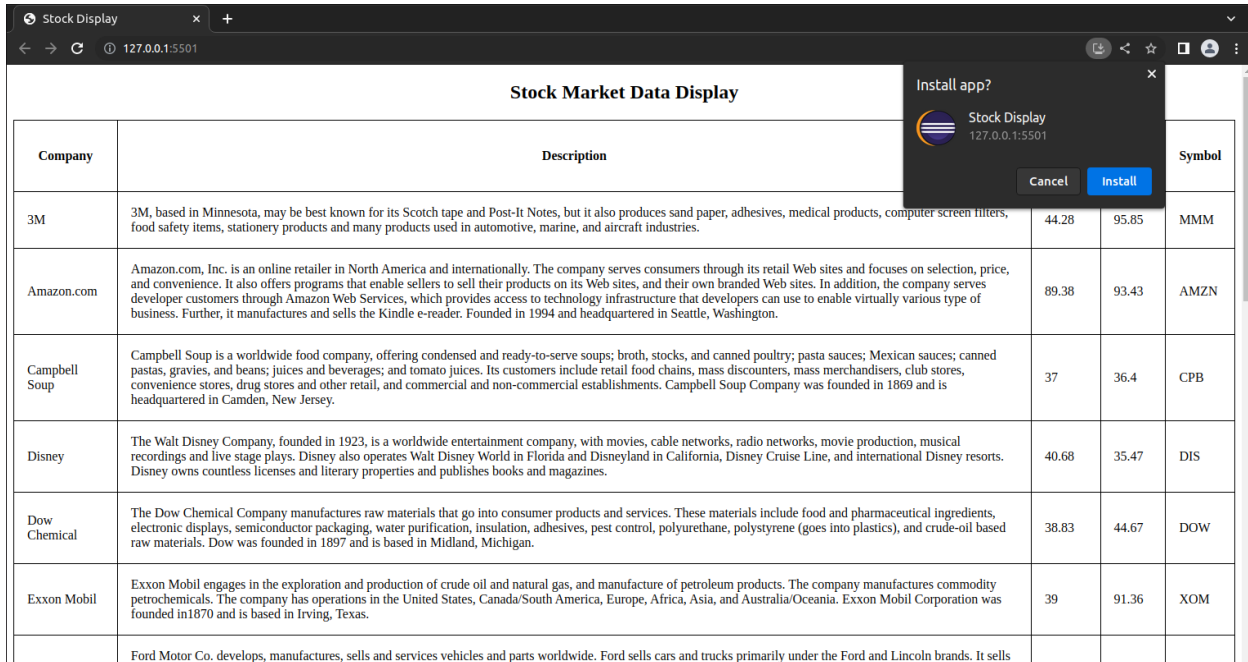
            }

            table += "</table>";

            // inserting the table into html
            document.getElementById("stock").innerHTML = table;
        })
    </script>
</body>
</html>

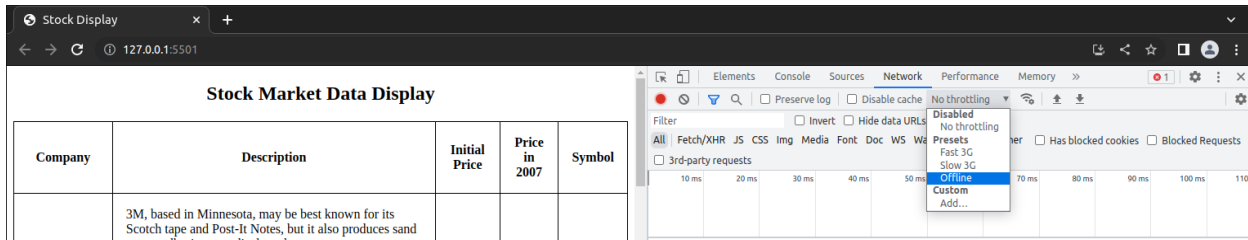
```


Output- (make sure to unregister the service worker from the previous programs in the applications tab of the inspect menu)



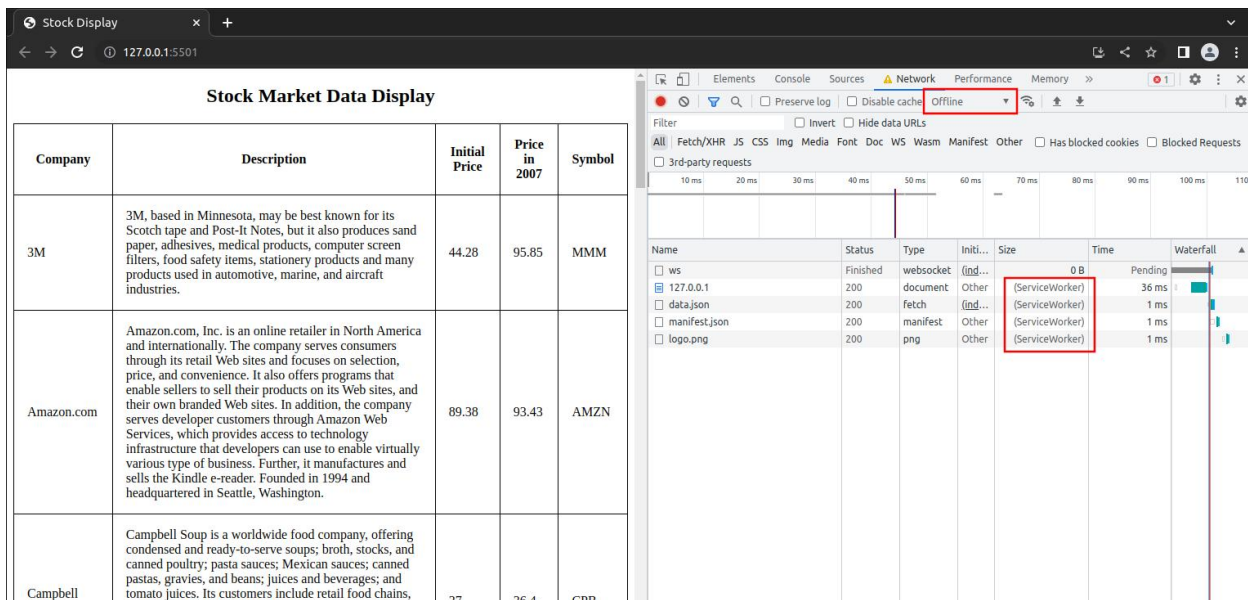
Stock Market Data Display					
Company	Description	Initial Price	Price in 2007	Symbol	
3M	3M, based in Minnesota, may be best known for its Scotch tape and Post-It Notes, but it also produces sand paper, adhesives, medical products, computer screen filters, food safety items, stationery products and many products used in automotive, marine, and aircraft industries.	44.28	95.85	MMM	
Amazon.com	Amazon.com, Inc. is an online retailer in North America and internationally. The company serves consumers through its retail Web sites and focuses on selection, price, and convenience. It also offers programs that enable sellers to sell their products on its Web sites, and their own branded Web sites. In addition, the company serves developer customers through Amazon Web Services, which provides access to technology infrastructure that developers can use to enable virtually various type of business. Further, it manufactures and sells the Kindle e-reader. Founded in 1994 and headquartered in Seattle, Washington.	89.38	93.43	AMZN	
Campbell Soup	Campbell Soup is a worldwide food company, offering condensed and ready-to-serve soups; broth, stocks, and canned poultry; pasta sauces; Mexican sauces; canned pastas, gravies, and beans; juices and beverages; and tomato juices. Its customers include retail food chains, mass discounters, mass merchandisers, club stores, convenience stores, drug stores and other retail, and commercial and non-commercial establishments. Campbell Soup Company was founded in 1869 and is headquartered in Camden, New Jersey.	37	36.4	CPB	
Disney	The Walt Disney Company, founded in 1923, is a worldwide entertainment company, with movies, cable networks, radio networks, movie production, musical recordings and live stage plays. Disney also operates Walt Disney World in Florida and Disneyland in California, Disney Cruise Line, and international Disney resorts. Disney owns countless licenses and literary properties and publishes books and magazines.	40.68	35.47	DIS	
Dow Chemical	The Dow Chemical Company manufactures raw materials that go into consumer products and services. These materials include food and pharmaceutical ingredients, electronic displays, semiconductor packaging, water purification, insulation, adhesives, pest control, polyurethane, polystyrene (goes into plastics), and crude-oil based raw materials. Dow was founded in 1897 and is based in Midland, Michigan.	38.83	44.67	DOW	
Exxon Mobil	Exxon Mobil engages in the exploration and production of crude oil and natural gas, and manufacture of petroleum products. The company manufactures commodity petrochemicals. The company has operations in the United States, Canada/South America, Europe, Africa, Asia, and Australia/Oceania. Exxon Mobil Corporation was founded in 1870 and is based in Irving, Texas.	39	91.36	XOM	
Ford Motor Co.	Ford Motor Co. develops, manufactures, sells and services vehicles and parts worldwide. Ford sells cars and trucks primarily under the Ford and Lincoln brands. It sells				

Change the network mode to offline



Stock Market Data Display					
Company	Description	Initial Price	Price in 2007	Symbol	
3M	3M, based in Minnesota, may be best known for its Scotch tape and Post-It Notes, but it also produces sand paper, adhesives, medical products, computer screen filters, food safety items, stationery products and many products used in automotive, marine, and aircraft industries.	44.28	95.85	MMM	
Amazon.com	Amazon.com, Inc. is an online retailer in North America and internationally. The company serves consumers through its retail Web sites and focuses on selection, price, and convenience. It also offers programs that enable sellers to sell their products on its Web sites, and their own branded Web sites. In addition, the company serves developer customers through Amazon Web Services, which provides access to technology infrastructure that developers can use to enable virtually various type of business. Further, it manufactures and sells the Kindle e-reader. Founded in 1994 and headquartered in Seattle, Washington.	89.38	93.43	AMZN	
Campbell Soup	Campbell Soup is a worldwide food company, offering condensed and ready-to-serve soups; broth, stocks, and canned poultry; pasta sauces; Mexican sauces; canned pastas, gravies, and beans; juices and beverages; and tomato juices. Its customers include retail food chains, mass discounters, mass merchandisers, club stores, convenience stores, drug stores and other retail, and commercial and non-commercial establishments. Campbell Soup Company was founded in 1869 and is headquartered in Camden, New Jersey.	37	36.4	CPB	

The website must work even when it's offline



Stock Market Data Display					
Company	Description	Initial Price	Price in 2007	Symbol	
3M	3M, based in Minnesota, may be best known for its Scotch tape and Post-It Notes, but it also produces sand paper, adhesives, medical products, computer screen filters, food safety items, stationery products and many products used in automotive, marine, and aircraft industries.	44.28	95.85	MMM	
Amazon.com	Amazon.com, Inc. is an online retailer in North America and internationally. The company serves consumers through its retail Web sites and focuses on selection, price, and convenience. It also offers programs that enable sellers to sell their products on its Web sites, and their own branded Web sites. In addition, the company serves developer customers through Amazon Web Services, which provides access to technology infrastructure that developers can use to enable virtually various type of business. Further, it manufactures and sells the Kindle e-reader. Founded in 1994 and headquartered in Seattle, Washington.	89.38	93.43	AMZN	
Campbell Soup	Campbell Soup is a worldwide food company, offering condensed and ready-to-serve soups; broth, stocks, and canned poultry; pasta sauces; Mexican sauces; canned pastas, gravies, and beans; juices and beverages; and tomato juices. Its customers include retail food chains, mass discounters, mass merchandisers, club stores, convenience stores, drug stores and other retail, and commercial and non-commercial establishments. Campbell Soup Company was founded in 1869 and is headquartered in Camden, New Jersey.	37	36.4	CPB	