EXPERIMENT-9

Name: Jayanaath S | Subject: Full Stack

Section: 23BCC-1 | Subject Code:23CSP-339

UID: 23BCC70022 | Date: 10-11-2025

- **Aim:**

  To deploy a React application using **Nginx** as a web server, enabling it to be served efficiently over HTTP.

- **Theory:**

  A React is a JavaScript library for building user interfaces. By default, React apps run on a development server using npm start. For production, the React app must be built into static files (HTML, CSS, JavaScript), which can be served by a web server.

  Nginx is a high-performance web server that can serve static content, reverse proxy requests, and handle load balancing. Serving a React app through Nginx improves performance, enables caching, and makes it accessible via standard HTTP ports (like 80 or 443).

  Key points:

  - React builds into a /build folder containing index.html and static assets.
  - Nginx serves index.html for all routes in single-page applications (SPA) using a try_files directive.
  - Production deployment ensures optimized, minified assets.

- **Procedure:**

  1. Build the React App

  - Navigate to your React project folder:

    ```
    cd my-react-app
    ```

  - Build the production version:

    ```
    npm run build
    ```

  - This creates a `build/` directory containing all static files.

  2. Install Nginx (if not installed)

  - For Ubuntu/Debian:

    ```
    sudo apt update
    sudo apt install nginx
    ```

3. Configure Nginx

- Open Nginx configuration file:

```
sudo nano /etc/nginx/sites-available/default
```

- Replace the server block with:

```
server {
    listen 80;
    server_name your_domain_or_IP;

    root /var/www/my-react-app/build;
    index index.html;

    location / {
        try_files $uri /index.html;
    }
}
```

`try_files $uri /index.html;` ensures React routing works properly.

4. Copy Build Files to Nginx Root

```
sudo cp -r build/* /var/www/my-react-app/
```

- Ensure proper permissions:

```
sudo chown -R www-data:www-data /var/www/my-react-app
```

5. Restart Nginx

```
sudo systemctl restart nginx
```

6. Test the Application

- Open a browser and visit http://your_domain_or_IP.
- The React app should load successfully

- **Result**

  The React application was successfully deployed and served using Nginx. The app is accessible over HTTP, with all routes handled correctly. Static assets like CSS, JS, and images are served efficiently.

- **Output:**