# Overview

This document intention is to check the possibility of Offline working of a website. By offline it means, user should be able to save data offline in case there is any network breakage and should be able to save the offline data to servers once network connection is back again.

## Describe the Proof of Concept “Solution”

This is for offline working of a web application.

Following are the aspects of the implementation.

* User is able to list all the data in offline mode (caching of GET request). To implement this, we have traversed all the pages of the list in online mode. Now we can see all the pages in offline mode also as data once retrieved is cached in browser.
* User is able to cache a add request too (caching of POST request). To implement this, application is using the storage of browser to cache and add the POST data and whenever system is online, there is a button to submit the saved offline data. Now, clicking on the button will submit the offline saved data.

To work with offline we have used Service Worker Module, an Angular technique. It will cache the data in browser cache.

For inserting the data offline we have used local storage (Indexed storage) which will store the data while offline and whenever system is online, data can be submitted.

All the rest Services are written in Spring Boot.

## Practical Details of Proof of Concept Use

* Context of Program: Offline working of a Web Application
* Planning and Development Process:
  + Implement Rest services for backend to store and retrieve the data.
  + Implement a Reactive Form in Angular 4.
  + Use the Service Worker technique to cache all **GET** request.
  + Use **localStorage** of browser (Indexed Storage) to implement caching of **POST** request as well.
* Skills Needed: Spring Boot (Rest Services), Angular 4.

## Challenges

* Service worker is not getting registered in Chrome but is working fine in Firefox.
* Cross Domain handling of requests.
* Future challenges: Session wise handling of GET and POST request, Implementing Queue of cached POST request, caching PUT and DELETE request as well.

## Steps

* Firstly, Install node modules, using “npm install”. This installs all the dependencies required by the Angular Application.
* Run “npm install –g @angular/cli”. This installs Angular’s Command Line Interface.
* To install and use Service Workers. Do the following steps:-

1. Run “npm install @angular/service-worker”. Install’s the service worker’s library.
2. Run “ng set apps.0.serviceWorker true”. Sets the CLI configuration of “serviceWorker” to true, enabling the use of Service Worker in our Application.
3. Configure the app.module, add the following code inside the imports.



* Run “npm install –p http-server”, this downloads a package which creates a server locally, in which our Angular Application will run.
* To run the application, run “ng build --prod”. This bundles the Angular Application for production and hence, can be deployed.
* Now run “http-server –p ’port number’ ” to run the Angular Application.

## Community Support

<https://angular.io/guide/service-worker-intro>

https://serviceworke.rs/request-deferrer.html