**Project: Whirlpool FSL and AI Accelerator**

**Description**: The idea is to club Einstein Artificial Intelligence with Field Service Lightning to implement the image recognition feature in order to boost the productivity of mobile workers and hence impacting the overall revenue of the company.

For instance, the whirlpool field service representatives are currently using the mobile app to see the allotted service appointments, see the direction of the destination, check in, check out and call reports. However, we can extend this power by introducing Einstein AI. So if a mobile worker is a newbie then he/she can click an image of a damaged part and get all its details, be it its installation guidelines or any other details, and hence saving on time of service.

Another example could be a case of a complicated equipment where in even an experienced resource might lack confidence in how to deal with it, leading to frustrated customers and may be multiple trips to resolve the issue.

In such scenarios image recognition feature implementation would play a very important role.

**Plan**:

**Sign up for an account** - Get an Einstein Platform Services account using below link:

<https://api.einstein.ai/signup>

Download and save its key.

**Install cURL** : Raise a genie and get it installed using below link:

<https://curl.haxx.se/download.html>

**Generate a token** - Each API call must contain a valid OAuth token in the request header. Use einstein\_platform.pem file to generate the token.

**Create a Remote Site Setting** - Before your Apex code can call the Einstein Vision endpoint, you must register the endpoint URL in the Remote Site Settings page.

**Prepare Data Set** - Collect images to build the data training set using either csv, tsv or json file. We need to add the labels as well.

**Train the Data Set** - Train the data set using cURL and note the model ID.

**Create classes** - Create the classes that create a web request and call the API endpoint.

**Create VF Page** - VF page from where the service resource will upload the image and call the classes to display the prediction results.

**Test the predictions.**