

EXCHANGE OF GEO-SPATIAL DATA THROUGH GEO-WEB

Abstract

Applications for the web, desktop, and mobile are made using the Arc-GIS server. Arc-GIS for server offers end users apps and services for managing, visualising, and analysing geographical data.

The proposed work deal with sharing of GIS data to any user using the provision of Arc-GIS server. It provides the facility of publishing Arc-map data in the global server and local server which facilitates user to share and access the Geo spatial data i.e vector data and Raster data.

Intelligent online maps offer a distinctive method of viewing complex data. They assist you in mining large datasets, uncovering hidden patterns, and pounding up seemingly unconnected facts. Using Arc-GIS server Online, data may be created on a desktop, transferred to the cloud, and shared.

Keywords: Arc-GIS Server, Raster Data, Vector Data.

Authors

Ballu Harish
Research Scientist
NRSC RRSC Hyderabad
Employee Code NR08270
harishballu111@gmail.com

Dr. R S Dwiwedi
Retd Group Director
NRSC Hyderabad
rsdwiwedi51@gmail.com

I. INTRODUCTION

The main server for Esri's Geographic Information System (GIS) is called ArcGIS Server. Create and manage GIS Web services, applications, and data using ArcGIS Server. ArcGIS Server is often installed either on-premises in the service-oriented architecture (SOA) of the enterprise, off-premises in a cloud computing environment, or both.

A complete and integrated server GIS is Arc-GIS server. It lets users to share tools, models, and maps with people inside and outside of their company in a way that works well for their workflows.

1. With Arc-GIS Server, user can:

- With real-time Geo-intelligence, publish quick, insightful Web maps that are suited to your audience, significantly enhancing business and resource decisions.
- Geographically enabling IT investments, reducing data and application redundancy, enhancing system setups, and consolidating enterprise systems.
- Manages geodata centrally, improving data security and integrity for the majority of key information assets.
- Simplify access to massive amounts of picture resources, significantly lowering the need for storage and data processing.
- Make Geographic Information System (GIS) technology available to your mobile workforce to improve resource and event management, asset monitoring, and field data gathering projects' accuracy and value.
- The client needs GIS in order to view a property on a map, analyse demographic data, and conduct a business analysis.

2. Using Arc-GIS server user can:

- **Create Web Mapping Application:** Constructing and overseeing browser-based tools for GIS data viewing, querying, and editing No prior programming experience is necessary. Both the Advanced and Standard editions of ArcGIS Server come with the programme.
- **Create Custom Web Applications:** Using JavaScript, Adobe Dreamweaver,.NET, or Java, create sophisticated interactive apps on top of ArcGIS Server.

3. Arc-GIS Server Supports a Wide Range of Applications:

- Arc-GIS Online: Using the Web to access and share data, maps, tools, and services.
- Mobile: Designing and overseeing mobile initiatives that function both while connected and when they are not in order to continue collecting and editing data. With the Advanced edition of ArcGIS Server, this is included in the box.
- Arc-GIS Desktop: Creating, enhancing, and utilising spatial analysis tools, map services, and GIS data.

- Arc-GIS Explorer is a tool for accessing, arranging, and displaying 2D and 3D GIS map services, tools for spatial analysis, and online content like Arc-GISSM. You can get this for free.
- Arc-GIS for AutoCAD: Provides access to vector, raster, and
- Image services for AutoCAD users. You can get this for free.

Who Uses Arc-GIS Server: Arc-GIS Server Delivers Centralized Geographic Intelligence to Anyone, Anywhere

- GIS experts: GIS experts manage and disseminate GIS content, such as maps, globes, imagery, and geo-processing models, across organisational boundaries to mobile and information workers, as well as Web users and application developers. They do this by using Arc-GIS Server.
- Database administrators: They utilise Arc-GIS Server to provide centralised, scalable geodata storage space, advance geodata security and dependability, and provide multiuser access and editing capabilities to Web, desktop, and mobile users.
- Application developers: Arc-GIS Server Application Programming Interfaces (APIs) are used by application developers to create and deploy sophisticated interactive Web mapping applications that can adhere to various geospatial, IT, and defence interoperability requirements.
- IT administrators: Arc-GIS server is used by IT administrators to automate work, modernise corporate processes, boost productivity, and manage GIS Web services and applications within a service-oriented architecture (SOA).
- Information workers: Information professionals employ Arc-GIS Server apps and services to generate reports and analytic maps, access authoritative data and processes, and enhance workflows and customer service.
- Mobile workers: Arc-GIS Server is used by mobile workers to access and use mobile maps, track the location of assets, and gather, modify, and update GIS data.
- Web citizens: Web users utilise Arc-GIS Server apps and services to view public data, provide their own maps, share comments, images, and other information, and take part in local, regional, and international decision-making.

4. Arc-GIS Server Extensions:

- It is used for adding More Capabilities and Share Your Desktop Models and Tools with the Extensions.
- 3D: examining what can be observed from various observation sites, subsurface features, and atmospheric occurrences, and visualising the best location for a facility.

- **Geostatistical:** Create statistically sound surfaces and more accurate forecasts by sharing model subsurface features, scientific models, and surface exploration approaches.
- **Network:** Create a realistic network model that includes features like multipoint routing, drive times, service area definition, best routes, and analysis of the closest facilities.
- **Spatial:** Using integrated raster and vector analysis, spatially generate new surfaces, forecast expected outcomes, identify model alternatives, and determine appropriate locations.
- **Image:** Enable on-the-fly processing and active mosaicking for a lot of photos. Make your imagery's complete information content accessible.
- **Data Interoperability:** Use spatial extract, transform, and load (ETL) capabilities to access and comprehend hundreds of different data formats.
- **Schematics:** Creating, viewing, and modifying diagrams that are kept in a geodatabase, as well as updating and sharing many network diagrams in a single layer, are all examples of schematics.
- **Job tracking:** administer project tracking and workflow management solutions to advance the proficiency of any multiuser GIS project.
- **Geo-portal:** Quick, dependable access to a variety of dynamic geospatial resources is provided via geoportals, which also support OGC specification standards, ISO 19139, FGDC metadata, and Dublin Core metadata.

5. ArcGIS Server Key Features

- **Data management:** Offers a framework and tools for handling massive spatial datasets in an RDBMS such as IBM® DB2®, Oracle®, IBM Informix®, PostgreSQL, and Microsoft SQL Server® as well as geo-data services for data replication, extraction, and management.
- **GIS web services:** Supports a variety of Web services, including geoprocessing, WMS, KML, WCS, WFS, and WFS-T.
- **Mapping:** Include tools for building feature-rich browser-based Web mapping applications.
- **Spatial analysis:** Supports scripts, models, and tools for server-based analysis and geoprocessing, including raster, vector, and network analytics.
- **Publishing to clients:** supports several different clients, such as ArcGIS Explorer, ArcGIS Desktop, AutoCAD, and browser-based and mobile apps.

- Image management: Supports a full image management system for delivering massive amounts of images that can be viewed in clients for mobile, desktop, the Web, and imaging.
- Web application functionality: Enables the building of Web mapping apps using a straightforward wizard with no coding needed.
- Application developer tools: Provides a full development environment for the Java platform and the Microsoft.NET Framework, as well as APIs for Adobe Flex, JavaScript, and Microsoft Silverlight for building sophisticated interactive applications.
- Spatial web editing functionality: Editing of map characteristics and features in web applications is supported by the spatial web editing functionality.
- Advanced Spatial Analysis: Supports complex geographic and statistical analyses using the free to use Geostatistical, 3D, Network, and Spatial extensions.
- Mobile GIS application functionality: Offers customizable mobile applications that let a mobile workforce dynamically query and update data as well as access server data remotely. These applications are managed by Arc-GIS Server, coupled with GPS, and easily fit into enterprise IT settings.
- Mobile application developer tools: Offers software developer kits (SDKs) and other management and setup tools for managing and configuring unique mobile applications.

II. LITERATURE REVIEW

All facets of disaster management—response, preparation, recovery, and mitigation—require geographic data and tools. Recently, a credible alternative to traditional information from mapping agencies and companies has emerged: geographic information created by leisure individuals, often known as volunteered geographic information.

Since volunteered information is unassured and without any of the guarantees that promote trust in data that is formally created, data quality is a key challenge. When there is an emergency, time is of the essence, and the dangers of using information that has been provided voluntarily are frequently exceeded by the advantages of doing so (Goodchild et al., 2010). Individuals are now able to build global-patch works of geographic information thanks to the rise in interest in gathering, creating, and disseminating geographic information on the web through sites like Wikimedia and open-street map (Michael F, Goodchild 2007).

Using the Geospatial Web, connecting with like-minded people in productive groups, and communicating across organisations. Geo-browsers like Google Earth and NASA World Wind are the enabling technology for the Geospatial Web. The development and consumption of media products are changing as a result of these three-dimensional platforms. They not only connect users with shared interests and browsing habits, but they also make the

geographic distribution of Web services and resources visible (Scharl, A. and Tochtermann, K. 2007).

Geospatial information and tools have the ability to save lives, reduce property damage, and lower disaster response costs in all facets of emergency management. Any technology's success depends equally on the human systems it is integrated into and the technology itself (National Research Council. 2007).

The development of the service-based GIS model is a result of the Open GIS Consortium's (OGC) concerted efforts to provide funding for the consensus-based creation of interoperable GIS Web service interfaces, as well as improvements in general Web service technology. For use in a range of client applications, it is possible to dynamically collect data from numerous GIS Web services.

Geographical information can be published and disseminated within organisations, across organisational boundaries, and to the general public using the server-based platform provided by ESRI. The main server GIS software from ESRI is Arc-GIS Server. It is a full-featured, integrated server-based GIS that offers end-user apps and services for managing, visualising, and analysing spatial data right out of the box.(The International Archives of the Spatial Information Sciences, Photogrammetry, Remote Sensing and LIU Laixing,2008).

By creating unique web services, Arc-GIS Server, the online application solution from ESRI, can be improved. Arc-GIS Desktop, rich Internet applications like the Arc-GIS API for Microsoft Silver light/WPF, and web applications like the web Application Development Framework (ADF) for Java / Microsoft.NET and Server Object Extensions offered by ESRI are just a few examples of client applications and developer frameworks and APIs that can be used to access Arc-GIS server services. Additionally, Arc-GIS Server offers a broad range of standards-based APIs that enable practically every client application to interact with and utilise Arc-GIS Server's mapping, spatial analysis, and data management features (12th Esri India User Conference, 2011)

Arc-GIS is a piece of general-purpose GIS software created by ESRI. It is a comprehensive piece of software for creating functional GIS. A set of out-of-the-box applications for creating, manipulating, editing, analysing mapping, and disseminating geographic information; a geographic information form for modelling aspects of the real world; and a collection of web services that provide content and capabilities to networked software clients are the four software parts that make up Arc-GIS. The Arc-GIS software system can be installed in parts on laptops, smartphones, desktop PCs, and servers. From the perspective of the end user, Arc-GIS offers a very broad range of capability that is condensed into a standard collection of menu-driven GIS apps that execute essential geographic workflows. The applications deal with geographic data formation (Encyclopedia of GIS, David J. Maguire, 2008)

Asia is now known for its biodiversity shifts, where the world's greatest biodiversity loss is seen (Yahara, 2009). Asia has developed a network of biodiversity research due to the availability of study areas for conducting integrative elucidation. Japan, Indonesia, and Cambodia are home to significant biodiversity observations.

Rigid observations are being made at these locations in the cooperation of the public, scientific, and citizen sectors. In order to define interconnections and interdependencies in research and to increase global responsiveness, coordinated tactics created by consuming information and integrating data are essential. The goal of the research in this area is to expand the Geographic Information System (GIS)-based server system database in order to enhance internal workflows at field interpretation and to more quickly share updated information on biodiversity surveillance data. The study concerns in the biodiversity information via GIS classification are first presented in this work. Second, a proposal is made to use a GIS portal to integrate data seamlessly across the entire biodiversity information system. Third, the Arc-GIS server and Microsoft Visual Studio 2008 are combined to create the GIS server structure programme, which serves as the system's core. Finally, a case study of a biodiversity observation field is currently constructed for sharing, producing, and exploiting biodiversity information, at the existing research outcomes, to verify the proficiency of the GIS server system application. (AASRI Conference on Computational Intelligence and Bioinformatics, Ibrahim Djameluddin ,2012)

1. Statement of the problem

- Sharing Geo-Spatial data is vital and difficult task using Current technology.
- Sharing information across national government and among users is not easier with in stipulated time.
- Existing hub facility for accessing geo-spatial data is not easier task.

2. Project objectives

The objectives of the work is

- Accessing the Geo-Spatial data from ESRI Arc-GIS server using different services like Map service, Feature service, Image service, etc..
- Displaying the details of Geo-Spatial data using the service Popup Template.
- Publishing the Geospatial data to Arc-GIS server Environment.

3. Methodology

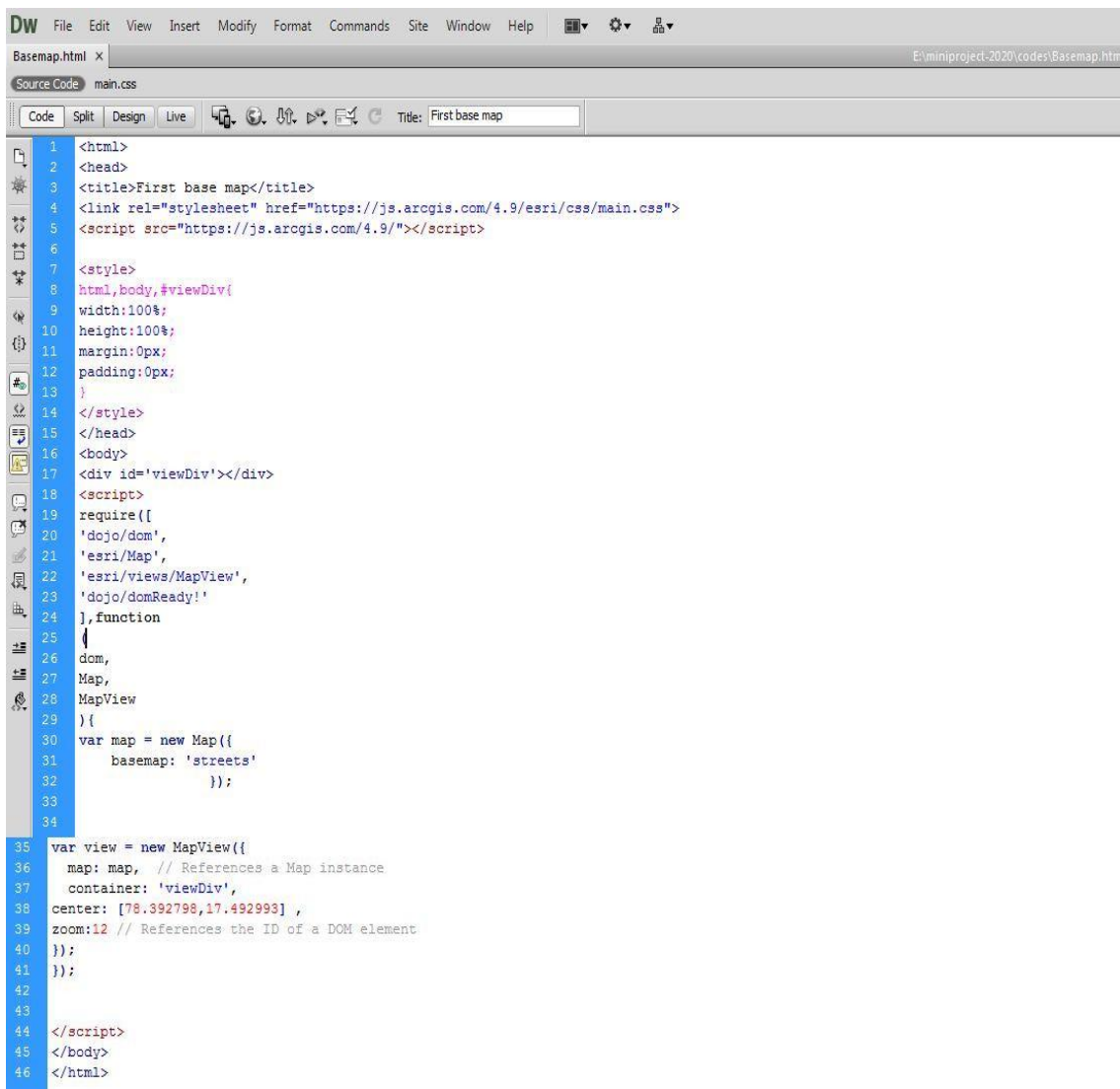
- using the website to insert the Base map
- To design the web pages using style-sheets of different techniques and using Dojo Toolkit for placing events.
- Creation of Map view environment for viewing of Geo-Spatial data using various services like Map service, Feature service.
- Creation of Rendering tools for customising vector data
- Generation of Query Task for providing information of vector data using pop up template
- Placing the widgets for knowing the information of Geo-Spatial data.
- Providing various Toggles for different view of spatial data.
- Publishing Geo Spatial data in server.

4. Work flow of the project

Accessing the webpage for placing the Base map: We are utilizing the Base-map attribute to display the base map on the HTML page. Different values for the Base-Map attribute include topo, streets, satellite, hybrid, dark-gray, gray, national-geographic, etc. For the base-map property, we can choose any value based on our dependencies. After that, we can display the base map by using the Map-view property. Additionally, the map-view has features like animation, background, container, center, zoom, navigation, and more. We can choose the necessary properties based on our requirements.

It is available under “**esri/Map**” for displaying the map we need view it is available under “ **esri/views/MapView**”.

Sample code for Base-Map and Map-view:



```
1 <html>
2 <head>
3 <title>First base map</title>
4 <link rel="stylesheet" href="https://js.arcgis.com/4.9/esri/css/main.css">
5 <script src="https://js.arcgis.com/4.9/"></script>
6
7 <style>
8 html,body,#viewDiv{
9 width:100%;
10 height:100%;
11 margin:0px;
12 padding:0px;
13 }
14 </style>
15 </head>
16 <body>
17 <div id='viewDiv'></div>
18 <script>
19 require([
20 'dojo/dom',
21 'esri/Map',
22 'esri/views/MapView',
23 'dojo/domReady!'
24 ],function
25 |
26 dom,
27 Map,
28 MapView
29 ){
30 var map = new Map({
31 basemap: 'streets'
32 });
33
34
35 var view = new MapView({
36 map: map, // References a Map instance
37 container: 'viewDiv',
38 center: [78.392798,17.492993] ,
39 zoom:12 // References the ID of a DOM element
40 });
41 });
42
43
44 </script>
45 </body>
46 </html>
```

Figure 1: Code1: Base-map code.

Figure 1 in the results section displays the above code's output. The Center property receives the values for Latitude and Longitude. I've provided the latitude and longitude values for a region of Hyderabad in this code (code-1). As a result, their work will become more focused.

5. Designing of the web pages using style-sheets and Dojo Toolkit for placing events: Using CSS, web page design is accomplished. The term CSS stands for Cascading Style Sheets. CSS explains how HTML elements should appear on a screen, in print, or in other media. A lot of work is saved via CSS. It can simultaneously control the layout of several web pages.

Designing of style sheets is done using three different ways:

- Inline style sheet
- Internal style sheet
- External style sheet

Inline style sheet: Inline CSS is a type of style sheet that includes CSS properties in the body of an element that is attached to it. Using the style attribute, this form of style is specific within an HTML tag.

Internal style sheet: When only one HTML document needs to be styled differently, this can be utilized. The head section of the HTML file contains the CSS rule, which means that the CSS is set within the HTML file.

External style sheet: With the use of the tag attribute, external CSS contains divided CSS files that only contain style properties (for example, class, id, header,... etc.). CSS property should be linked to the HTML document using a link tag and written in a separate file with the extension ".css."

This means that for each element, style can be put only once and that will be useful across web pages.

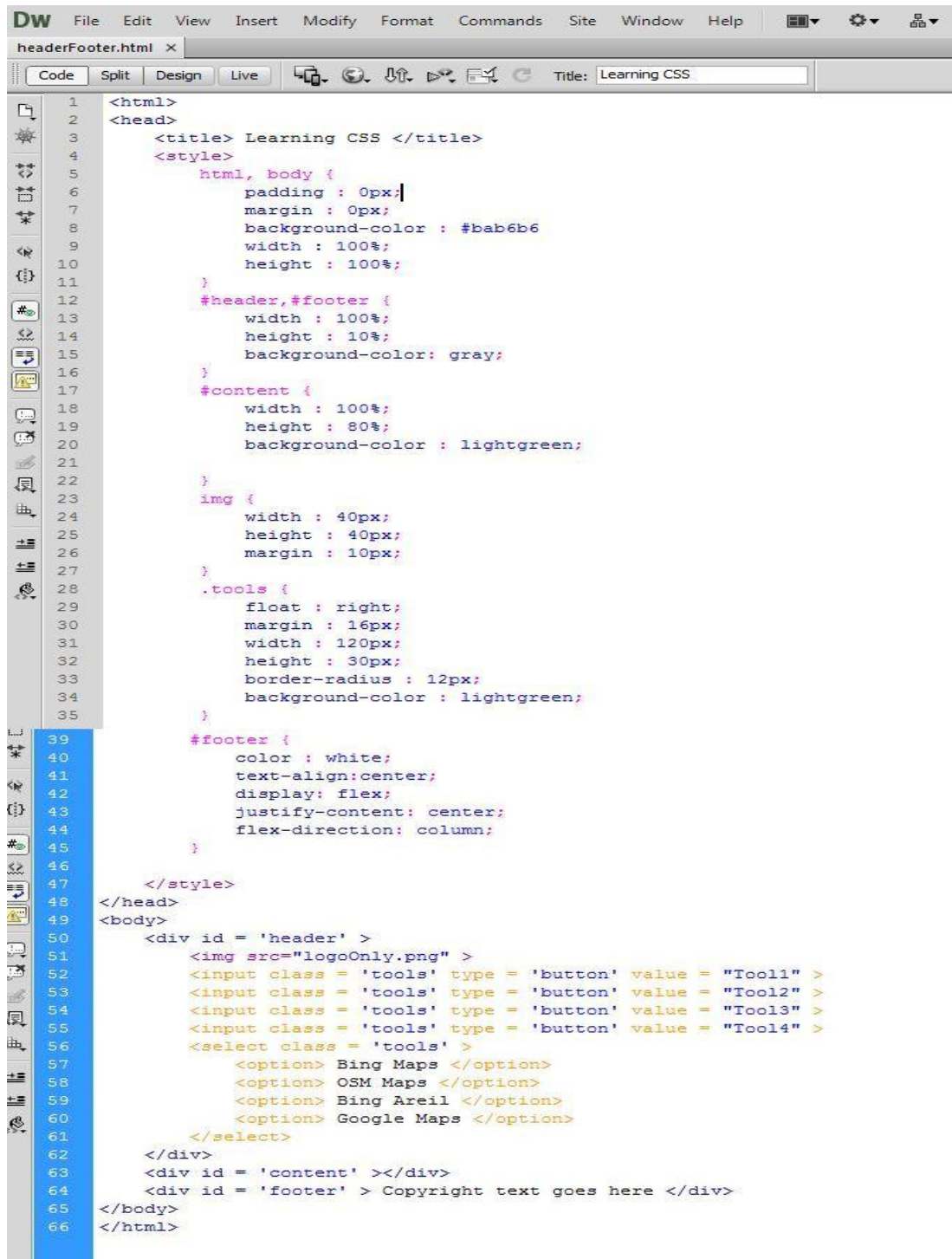
The manual entry of code by the programmer is required by the CSS ideas, and it takes longer when customized. Therefore, in order to customize in less time, we switch to the Dojo toolkit, which comes with built-in style code. By using it, we can reduce the amount of programming required to style the web pages.

Dojo Tool-Kit: The Dojo toolkit includes a Java-Script library that is designed to make it simple to build cross-platform JavaScript and Ajax-based websites. By employing the web standards themselves as the platform, it attempts to cut down on time and scale the advancement approach.

It can be used in two different ways

- Using CDN i.e by downloading the dojo in the local system it can be used in our personnel system directly.
- Using its API it can be accessed from the web environment of Dojo.

Sample code for CSS:



```
1 <html>
2 <head>
3 <title> Learning CSS </title>
4 <style>
5   html, body {
6     padding : 0px;
7     margin : 0px;
8     background-color : #bab6b6
9     width : 100%;
10    height : 100%;
11  }
12  #header,#footer {
13    width : 100%;
14    height : 10%;
15    background-color: gray;
16  }
17  #content {
18    width : 100%;
19    height : 80%;
20    background-color : lightgreen;
21  }
22  }
23  img {
24    width : 40px;
25    height : 40px;
26    margin : 10px;
27  }
28  .tools {
29    float : right;
30    margin : 16px;
31    width : 120px;
32    height : 30px;
33    border-radius : 12px;
34    background-color : lightgreen;
35  }
36  }
37  #footer {
38    color : white;
39    text-align:center;
40    display: flex;
41    justify-content: center;
42    flex-direction: column;
43  }
44  }
45  }
46 </style>
47 </head>
48 <body>
49 <div id = 'header' >
50 
51 <input class = 'tools' type = 'button' value = "Tool1" >
52 <input class = 'tools' type = 'button' value = "Tool2" >
53 <input class = 'tools' type = 'button' value = "Tool3" >
54 <input class = 'tools' type = 'button' value = "Tool4" >
55 <select class = 'tools' >
56 <option> Bing Maps </option>
57 <option> OSM Maps </option>
58 <option> Bing Areil </option>
59 <option> Google Maps </option>
60 </select>
61 </div>
62 <div id = 'content' ></div>
63 <div id = 'footer' > Copyright text goes here </div>
64 </body>
65 </html>
```

Figure 2: Code2: CSS code.

The output of the code is shown in result section with figure 2. This code (code-2) completely deals with styling of web page .various properties of css is used in the code on the dependency of output.

Sample code of Dojo:

```
<html>
<head>
  <title> DOJO Title Pane </title>
  <script src="https://ajax.googleapis.com/ajax/libs/dojo/1.13.0/dojo/dojo.js"></script>
  <link rel="stylesheet" href="http://ajax.googleapis.com/ajax/libs/dojo/1.13.0/dijit/themes/claro/claro.css">
  <style>

    html,body,#layContainer {
      width:100%;
      height: 100%;
      margin:0px;
      padding:0px;
    }
    #tpContainer {
      width : 300px;
      height : 200px;
      top:20px;
      right:20px;
      position:absolute;
      z-index : 9999;
    }
    #cpHeader,#cpFooter {
      height: 10%;
      background-color:gray;
    }
    #cpContent {
      height : 80%;
      background-color : lightBlue;
    }
    #cpLeft {
      width:300px;
      background-color:red;
    }
  </style>

  <div id='layContainer' data-dojo-type='dijit/layout/LayoutContainer' data-dojo-props="design:'headLine'" >
    <div id='cpHeader' data-dojo-type='dijit/layout/ContentPane' data-dojo-props="region:'top'" >
    </div>

    <div id='cpLeft' data-dojo-type='dijit/layout/ContentPane' data-dojo-props="region:'left'">
    </div>

    <div id='cpContent' data-dojo-type='dijit/layout/ContentPane' data-dojo-props="region:'center'">
      <div id='tpContainer' >
        <div id='tPane' data-dojo-type="dijit/TitlePane" data-dojo-props="title:'This is the title', open:false" >
          <div id='lyrName' data-dojo-type='dijit/form/TextBox' ></div>
          <div id='btnOK' data-dojo-type='dijit/form/Button' >OK</div>
        </div>
      </div>
    </div>
    <div id='cpFooter' data-dojo-type='dijit/layout/ContentPane' data-dojo-props="region:'bottom'">
    </div>

  </div>

  <script>
    require([
      'dojo/dom',
      'dojo/on',
      'dijit/registry',
      'dojo/parser',
      'dojo/domReady!'
    ], function(
      dom,
      on,
      registry,
      parser
    ){
      parser.parse().then( function() {
        /*
          on(dom.byId('btnOK'), 'click', function(){
            alert('Clicked');
          });
          */
          registry.byId('btnOK').on('click',function(){
            alert(dom.byId('lyrName').value);
          });
        });
      });
    </script>
  </body>
</html>
```

Figure 3: Code3: DOJO- code.

The output of the code is shown in result section with figure 3. This code (code-3) completely deals with styling of web page using the available tools of Dojo toolkit. Here in the code some of the properties of Dojo like Title-Pane, Content-Pane, Text box, Buttons have been used. If the creation of the mentioned properties need to be done .Without Dojo concept we need manually write hundreds line of code where as above programme contains only ninty lines of code i.e Three fourth of the code is reduced in the programme.

In the above code (code-3) tags like script and link have been used which will contain the predefined code of Dojo because of the two tags only coding effort got reduced in the programme.

- 6. Displaying the Services in Map view environment:** Numerous services, including map services, feature services, image services, geoprocessing services, and geocoding services, are easily accessible.

Map Services are the equivalent of an Arc-Map.mxd file.

One map service is made up of many Feature services.

A feature service is one that focuses on a single feature, such as a point, line, or polygon.

Raster data, such as topographical sheets and satellite images, are focused using image services.

For publishing tools that are already in existence or tools that users have just generated, geoprocessing services are used.

Geo-Coding services provide addresses when x, y, and z are known.

- 7. Map Image Layer and Feature Layer:** It is available under

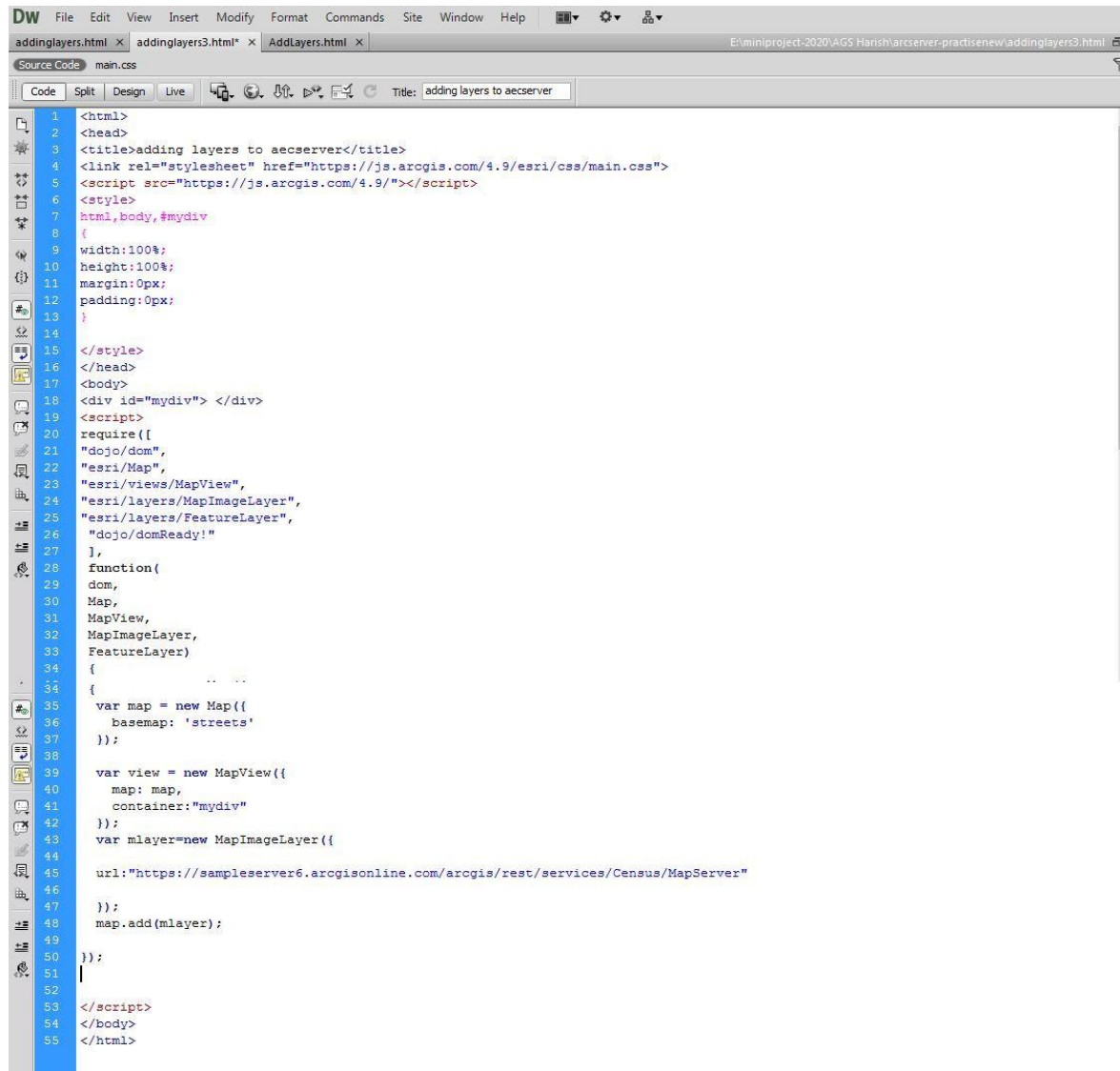
- "esri/layers/MapImageLayer" and
- "esri/ layers/FeatureLayer " API reference.

The ability to examine data from defined sublayers in a map service and export images rather than features is provided by the map image layer. Based on a demand that comprises a bounding box, LOD (level of detail), dpi, spatial reference, and other variables, map service images are quickly created on the server. The complete map extent defined in the produced image.

A Map Service or Feature Service, Arc-GIS Online or Arc-GIS Enterprise portal entries, or a variety of client-side descriptions can all be used to produce a single layer known as a Feature Layer. The layer may be spatial or non-spatial (such as a table) (have geographic features) Each discrete feature in a spatial layer has a geometry that enables it to be shown as a graphic with a spatial framework in either a 2D map view or a 3D scene view. Additionally, features have data attributes that give additional details about the features in real life that it represents. These attributes can be examined in pop-up

windows and are utilized to render the layer. For the purpose of visualizing data in a spatial context, feature layers can be evaluated, queried, and rendered.

Sample code of Map Image layer:



```
1 <html>
2 <head>
3 <title>adding layers to aecserver</title>
4 <link rel="stylesheet" href="https://js.arcgis.com/4.9/esri/css/main.css">
5 <script src="https://js.arcgis.com/4.9/"></script>
6 <style>
7 html,body,#mydiv
8 {
9 width:100%;
10 height:100%;
11 margin:0px;
12 padding:0px;
13 }
14
15 </style>
16 </head>
17 <body>
18 <div id="mydiv"> </div>
19 <script>
20 require([
21 "dojo/dom",
22 "esri/Map",
23 "esri/views/MapView",
24 "esri/layers/MapImageLayer",
25 "esri/layers/FeatureLayer",
26 "dojo/domReady!"
27 ],
28 function(
29 dom,
30 Map,
31 MapView,
32 MapImageLayer,
33 FeatureLayer)
34 {
35
36
37
38
39
40
41
42
43
44
45 url:"https://sampleserver6.arcgisonline.com/arcgis/rest/services/Census/MapServer"
46
47
48 map.add(mlayer);
49
50
51
52
53 </script>
54 </body>
55 </html>
```

Figure4: Code4: Map Image layer code.

- The output of above code (code-4) is shown in result section with figure 4. This code completely deals with contains sample census data of the United States. It contains the information reg four different layers i.e.
- Census Block Points (0): It is of point type.
- Census Block Group (1): It is of polygon type.
- Detailed Counties (2): It is of polygon type.
- States (3): It is of polygon type.

By inspecting the each layer we can get the information regarding its parameters like Spatial Reference, Initial Extent, Full Extent, Image Format Types and etc...

It is a feature of the Map-Image and Feature layers. The Esri/layers sections can be used to find out the value for url. All the links are shown in the services section at <https://sampleserver6.arcgisonline.com/arcgis/rest/>. With a slight variation, the same URL value can be provided to a feature layer; for example, each feature layer's URL ends with 0, 1, 2, etc.

Sample code of Feature Layer:

```
<script>
  require([
    'dojo/dom',
    'esri/views/MapView',
    'esri/Map',
    'esri/layers/MapImageLayer',
    'esri/layers/FeatureLayer',
    'dojo/domReady!'
  ], function(
    dom,
    MapView,
    Map,
    MapImageLayer,
    FeatureLayer
  ){
    var map = new Map({
      basemap: 'streets'
    });

    var view = new MapView({
      map: map,
      container: 'viewDiv',
    });

    var fLayer = new FeatureLayer({
      url: 'https://sampleserver6.arcgisonline.com/arcgis/rest/services/Census/MapServer/3'
    });

    map.add(fLayer);
  });
</script>
```

Figure 5: Code5: Feature Layer code.

- The output of above code is shown in result section with figure 5. This code (code-5) completely deals with contains sample census data of the United States. It contains the information regarding Layer3 only.
- States (3): It is of polygon type.

By inspecting the layer we can get the information regarding its parameters like Spatial Reference, Initial Extent, Full Extent, Image Format Types and etc...

URL: It is a property of Map-Image layer and Feature layer. The value for url can be known from Esri/layers sections. All the url's are mentioned at <https://sampleserver6.arcgisonline.com/arcgis/rest/> in the services section.

Here in Feature layer we focus on individual layer where as in map service we can't find the individual layers.

- 8. Creation of rendering tools:** The rendering tool is available under “**esri/Renderers**” API reference. It supports Eleven various techniques. Apart of those techniques we are going to use “Simple-Renderer”.

Simple Renderer renders all features in a Layer with one Symbol. This renderer is used to simply visualize the site of geographic features. For example, layers signifying political borders, cities, buildings, etc. that are typically used for providing geographic perspective in an application may be defined with a single symbol per layer.

Simple-Renderer may only be used to craft visualizations for the below mentioned layer types:

- FeatureLayer
- SceneLayer
- MapImageLayer
- CSVLayer
- StreamLayer

Using Rendering we can create different symbols for point/line/polygon.

Under the simple renderer we have different things to discuss.

Simple Marker Symbol: It is used to denote points.

Simple Line Symbol: It is used to denote Lines.

Simple Fill Symbol: It is used to denote Areas.

Sample code of Simple Renderer:



```
rendering.html x
Source Code main.css
Code Split Design Live Title: Rendering concept
No syntax errors.
1 <html>
2 <head>
3 <title>Rendering concept</title>
4 <link rel="stylesheet" href="https://js.arcgis.com/4.9/esri/css/main.css">
5 <script src="https://js.arcgis.com/4.9/"></script>
6 <style>
7 html,body,#mydiv
8 {
9 width:100%;
10 height:100%;
11 margin:0px;
12 padding:0px;
13 }
14
15 </style>
16 </head>
17 <body>
18 <div id="mydiv"></div>
19 <script>
20 require([
21 "dojo/dom",
22 "esri/Map",
23 "esri/views/MapView",
24 "esri/renderers/SimpleRenderer",
25 "esri/symbols/SimpleMarkerSymbol",
26 "esri/symbols/SimpleLineSymbol",
27 "esri/layers/MapImageLayer",
28 "esri/layers/FeatureLayer",
29 "dojo/domReady!"
30 ],
31 );
32
33 var view = new MapView({
34 map: map,
35 container:"mydiv"
36 });
37 var ptrender=new SimpleRenderer({
38 symbol:new SimpleMarkerSymbol({
39 style: "square",
40 color: "blue",
41 size: "8px",
42 outline: {
43 color: [ 255, 255, 0 ],
44 width: 3
45 }
46 })
47 });
48 var lnrender=new SimpleRenderer({
49 symbol:{
50 type:"simple-line",
51 color: "red",
52 width: "2px",
53 style: "solid"
54 }
55 });
56 var polrender=new SimpleRenderer({
57 symbol:new SimpleFillSymbol({
58 color: [ 51,51, 204, 0.9 ],
59 style: "solid",
60 outline: { // autocasts as new SimpleLineSymbol()
61 color: "red",
62 width: 2
63 }
64 })
65 });
66
67 //var fLayer=new FeatureLayer({
68 url:"https://sampleserver6.arcgisonline.com/arcgis/rest/services/Hurricanes/MapServer/0",
69 renderer:ptrender
70 });
71 map.add(fLayer);
72
73 /*var fLayer=new FeatureLayer({
74 url:"https://sampleserver6.arcgisonline.com/arcgis/rest/services/Hurricanes/MapServer/1",
75 renderer:lnrender
76 });
77 map.add(fLayer);
78 var fLayer=new FeatureLayer({
79 url:"https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/2",
80 renderer:polrender
81 });*/
82 map.add(fLayer);
83
84 });
85 </script>
86 </body>
87 </html>
```

Figure 6: Code6: Rendering- code for customization of all the vector data using feature layer

The output of above code is shown in figure-6,7,8. In the above code (Code-6) the customization of vector data can be individually at a time as we had worked with feature layer. If we want to customize all the layers at a time we need to work with Map-Image-Layer i.e customization for all the layers point, line, polygon can be done at a time.

9. Generation of query task: It is available under: “esri/tasks/QueryTask” API reference.

It executes different types of queries on a layer. The most common method used is execute (), which execute the query as defined in the Query object that is send as a factor to the function.

QueryTask.execute() uses the promise that resolves to a feature-set, which contain the features in the layer that suit the Query.

With QueryTask, you can also gain the number of features that suit the query, the extent of features, related records or features associated with a feature.

QueryTask.execute () returns a promise that resolve a feature set which contains features in the layer that suit the query.

Sample code of Query Task:

```
52
53     var query = new Query();
54     query.where = "STATE_NAME = 'Washington' OR STATE_NAME = 'Montana'";
55     query.returnGeometry = true;
56     query.outFields = ['STATE_NAME', 'NAME'];
57
58     var qTask = new QueryTask({
59         url: 'https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/3'
60     });
61
62     qTask.execute(query).then(function(results){
63         var features = results.features;
64         console.log(features);
65     }, function(error){
66         console.log(error);
67     });
68
69 }
```

Figure7: Code7: Query task code for operations on a layer

The output of above code (code-7) is shown in Figure 9. In the Query Task execution one of the constraints is, Information about the complete feature layer (Total vector data) is available in output at single instant only. In-order to view the individual vector data using its object id we can know about its details. To avoid this constraint popup-template is used for getting the details of any vector data using a single click.

10. Popup-template: It is available under “esri/popuptemplate”.

For a Graphic or particular Layer, a Popup-Template defines and formats the content of a Popup. When a feature in the view is selected, a Popup-Template enables the user to access data from feature characteristics and data returned from Arcade terms.

The Popup-Template contains content and title properties that act as a template used to make over a feature's attributes into an HTML representation. The syntax {field-Name} or {expression/expression-Name} performs parameter exchange. The default performance on a graphic is to prove the view's popup after a click on the graphic. A Popup-Template is required for this default behaviour.

By clicking on it, it provides information on the feature that is presented. The output map's zoom level and clicking on it change how it behaves. The user can get value from feature characteristics of a specific feature in output using a popup template. It has a variety of qualities, including actions, material, and titles, among others. Depending on our needs, we must choose the ones that best meet those criteria. It offers customisation for the "contain" property.

Sample code for Pop-up Template:

```
40 var view = new MapView({
41     map: map,
42     container:"mydiv"
43 });
44
45 var pTemplate= new PopupTemplate({
46     title: "Population of us",
47     content: "{*}"
48     //content: "state_name:{state_name}<br> state_abbr:{state_abbr} <img width='50px' height='50px'
49     src='https://jntuh.ac.in/assets/uploads/e182b-0Y4A7358.JPG'>"
50 });
51
52 var fLayer=new FeatureLayer({
53     url:"https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/2",
54     popupTemplate:pTemplate
55 });
56 map.add(fLayer);
57 });
```

Figure 8: Code 8: Popup template for displaying the feature information at the instant.

The output of above code (code-8) is shown in Figure 10. Popup-template is used for getting the details of any vector data using a single click on the selected feature.

11. Placing the widgets: It is available under “esri/widgets” API reference. There are many widgets available under this API. But here only Layer-List, Legend, Locate have been used. Based on our dependency we can opt the widget which suits to our need.

Layer-List: It is available under "esri/widgets/LayerList" API.

The Layer-List widget provides a way to exhibit a file of layers, and switch on/off their visibility. The List-Item API provide access to each layer's property, allows the developer to organize actions related to the layer, and allows the developer to insert content to the item related to the layer.

To conceal the layers in the map from the widget Layer-List we must set the property of list on the desired layers to hide.

Legend: It is available under " esri/widgets/Legend " API.

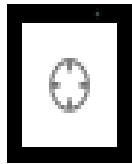
The Legend widget displays symbols and labels for layers in a map. Labels and their matching symbols depend on the value set in the renderer of the layer. The legend will only display layers and sub-layers that are evident in the view.

The legend automatically updates when

- the visibility of a sub-layer or layer changes
- a layer is new or isolated from the map
- The layer's renderer, title or opacity is changed
- the legend-enabled property is changed (set to false or true on the layer)

Locate: It is available under "esri/widgets/Locate" API.

Provides a simple widget that animate the view to the user's present location. The view rotates according to the route where the tracked device is heading towards. By default the widget looks like the following:



Sample code for widgets:

```
104         var miLayer = new MapImageLayer({
105             url: mapserver
106         });
107
108         var stLayer = new FeatureLayer({
109             url: "https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/2"
110         });
111         var cntLayer = new FeatureLayer({
112             url: "https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/3"
113         });
114         var hgtLayer = new FeatureLayer({
115             url: "https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/1"
116         });
117
118         map.layers.add(cntLayer);
119         map.layers.add(hgtLayer);
120         map.layers.add(stLayer);
121
122         map.layers.add(miLayer);
123         var lList = new LayerList({
124             view: view,
125             container: "LayerList"
126         });
127
128         var legend = new Legend({
129             view: view,
130             container: "legendDiv"
131         });
132
133         var locate = new Locate({
134             view: view
135         });
136
137         view.ui.add(locate, "bottom-left");
```

Figure9: Code 9: Widgets code for placing the widgets on the base map.

The output of above code (code-9) is shown in Figure 11,12, 13. Layer List code is used for appending the layers to base map mentioned in source code. Legend code is used for mentioning the information about features. Locate is used for navigating to the existing location of user.

12. Providing various Toggles: The Toggle is available under "esri/widgets" API reference. Here we are discussing about Basemap-Gallery. The Base map gallery is available under "esri/widgets/BasemapGallery".

The Base-map-Gallery widget will show the set of images representing base-maps from ArcGIS.com or a user-defined image services .When a new base-map is selected from the Base-map-Gallery, the map's base-map layers are removed and replaced with the base-map layers of the associated base-map selected in the gallery.

Sample code for Toggle:

```
44     parser.parse().then(function(){
45
46         var map = new Map({
47             basemap: 'streets'
48         });
49
50         var view = new MapView({
51             map: map,
52             container: 'viewDiv',
53             zoom : 4,
54             center : [77.461646,21.698939]
55         });
56
57         var bGallery = new BasemapGallery({
58             view: view,
59             container : 'bMapGallery'
60         });
61         |
62         view.ui.add(bGallery, "top-right");
63
64     });
65
66 });
67
```

Figure10: Code 10: Base Map Gallery code.

The output of above code (code-10) is explained in figure 14.

It will display all the maps at a time we can select any of them. By default Eleven different types of maps are available we can choose any of the map from them. The Base map Gallery is available under “Dojo” technique. Whenever the form gets loaded by default the BaseMapGallery will be under false mode, By clicking on the Dojo technique we can see all the available maps.

13. Publishing Geo Spatial data: Publishing the Geo Spatial data is important task as it facilitates the user work in sharing the data to destination in easy way. The publishing of

Data can be done in two ways

- It can be published in local server (Local system).
- It can be published in Global server.

Once the data is published it gets ready with all the supporting documents, i.e If a Geo referenced image is going to publish it gets shared to server along with its supporting documents i.e RRD file. In case of publishing we can publish the Images,vector data, Tools of Arc-GIS,Topo sheet data to the server system.

The output of above technique is discussed in **figure 15(Results)** section.

Results:

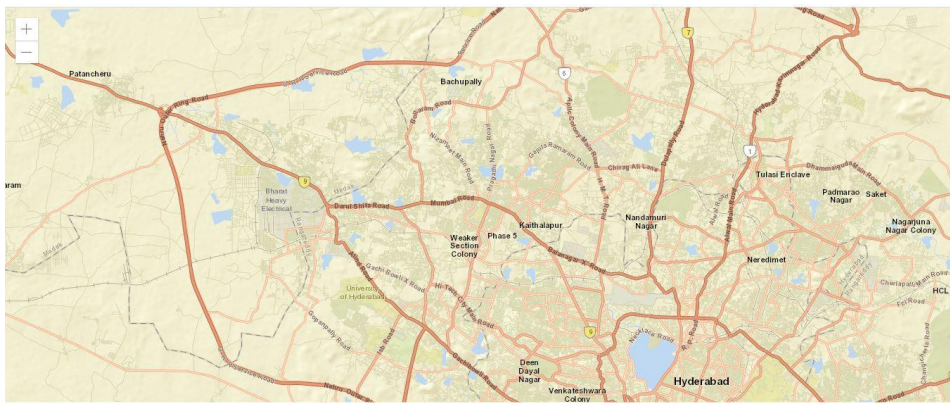


Figure 11: Base-Map is shown by focus on Hyderabad region.

Figure 1 up top shows an example base map that is displayed utilizing the Arc-GIS server API setup. The view set here in the aforementioned code was "streets." Alternative values, such as hybrid, satellite, and so forth, can also be utilized. Just the map is being shown here.



Figure12: Style-sheets provided for designing of web page.

In the above figure (figure2) a web page is designed and made customized by placing the events in it using css techniques a part of customization buttons, drop down also used in the output. The styles that be provided common or different for the view and events in the page.



Figure13: Styling of web page using Dojo Tool kit

Using the Style-sheets a webpage can be designed in an attractive way. It means customization of designing can also be done Dojo Tool kit. Using Dojo coding effort can be reduced for a programmer. Instead he can use the default code that was given with dojo.

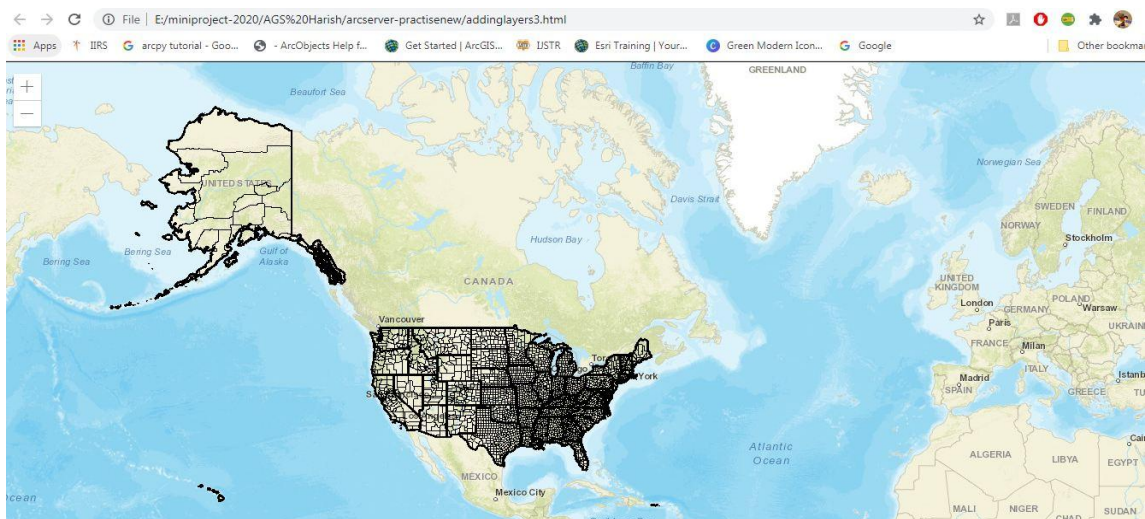


Figure 4: Map Image layer containing 3 layers of sample census data .

Using the Map Image layer, the above figure (figure 4) shows all the vector data layers at once. The parent layer of the feature layer is supposedly the map image layer. Three distinct layers, including point data, line data, and polygon data, are present in the output seen above.

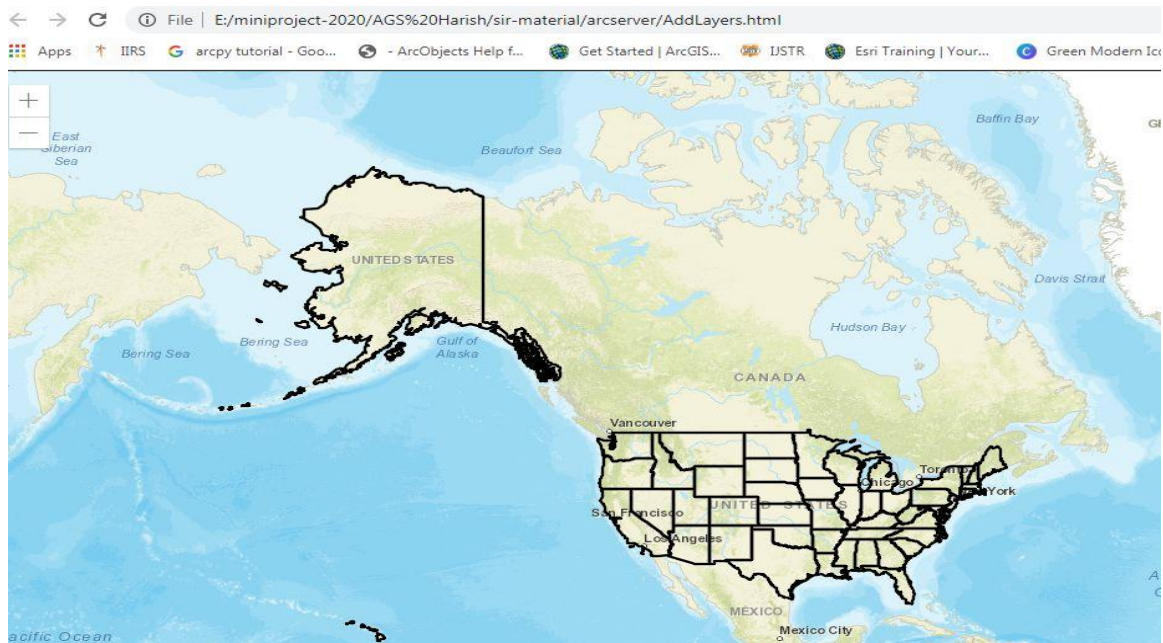


Figure 15: Feature Layer containing Layer 3 of sample census data.

The polygon layer of census data is depicted in the previous picture (figure 5). The feature layer can also be used to represent additional layers, such as line and point data. A single layer at a time is represented by a feature layer.

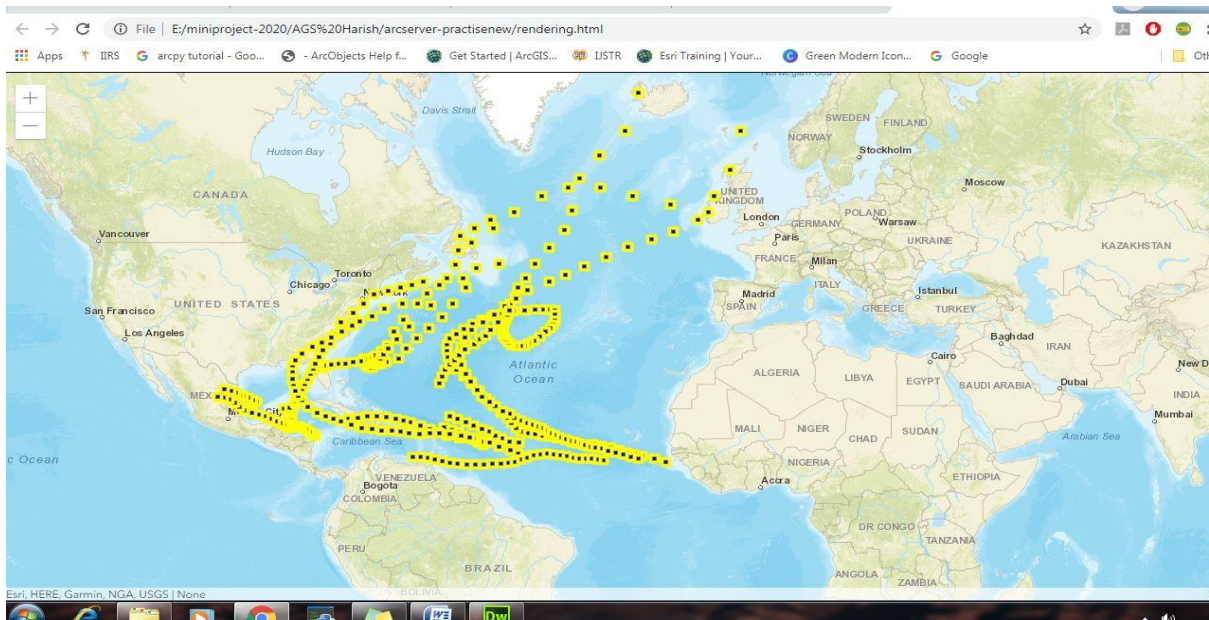


Figure 16: Customization of Point data using Simple Marker Symbol.

Simple-Marker-Symbol is used for rendering 2D Point geometries with a simple shape and colour in either a Map-View or a Scene-View.

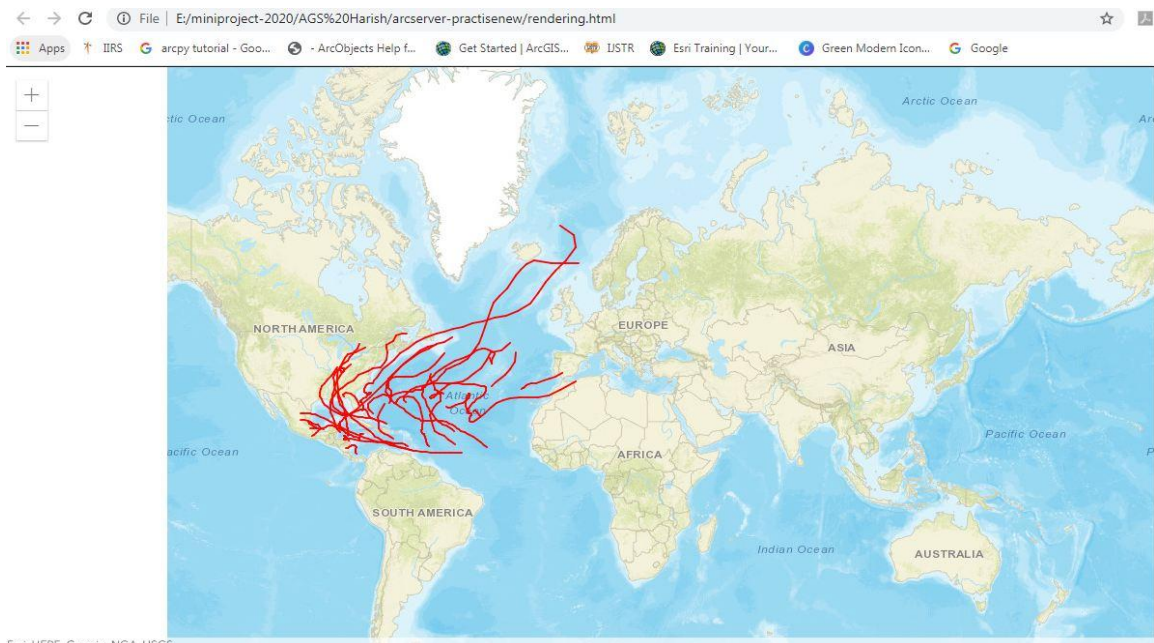


Figure17: Customization of Line data using Simple Line Symbol.

Simple-Line-Symbol may also be used to symbolize 2D polyline features in a Scene-View.

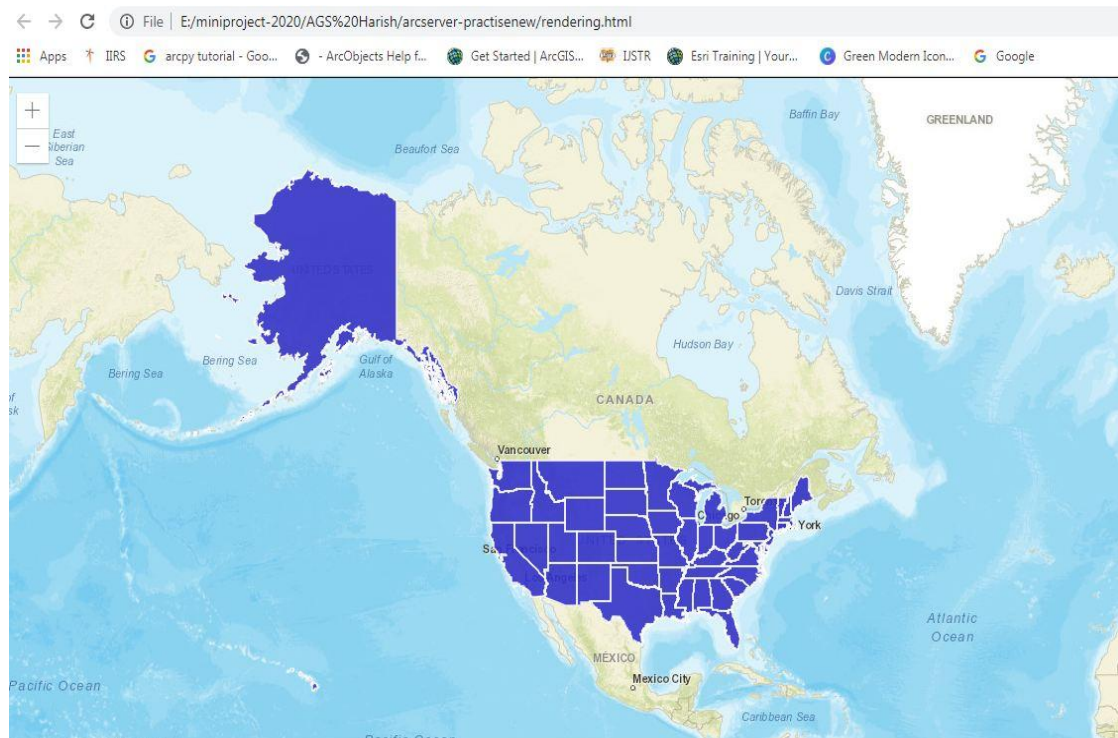


Figure 18: Customization of Polygon data using Simple Fill Symbol.

Simple Fill Symbol is used for rendering 2D polygons in either a Map-View or a Scene-View. It can be filled with a solid color, or a pattern. Simple-Fill-Symbols may be applied to polygon features in a Feature-Layer or an individual Graphic

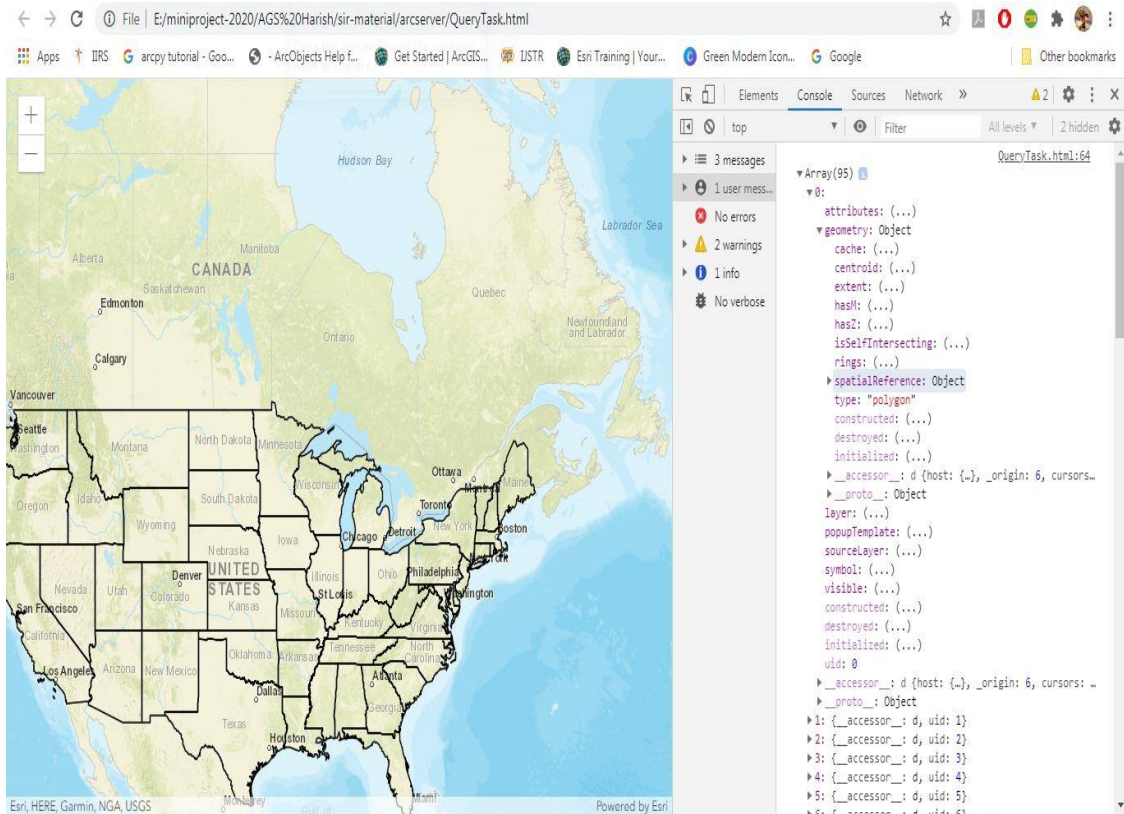


Figure 19: Query Task execution on the feature layer.

By looking the above output we can know the output of query task. Total information about feature layer is displayed, It can be known to user about total number of features present in the existing layer.

By navigating to developer tools → console section → using the array we can know number of features present in existing layer.

By clicking on any layer we can know its details like geometry, spatial reference, type and so-on etc...

Here knowing the object id and navigating is bit difficult to overcome this issue concept of pop-up template is used.

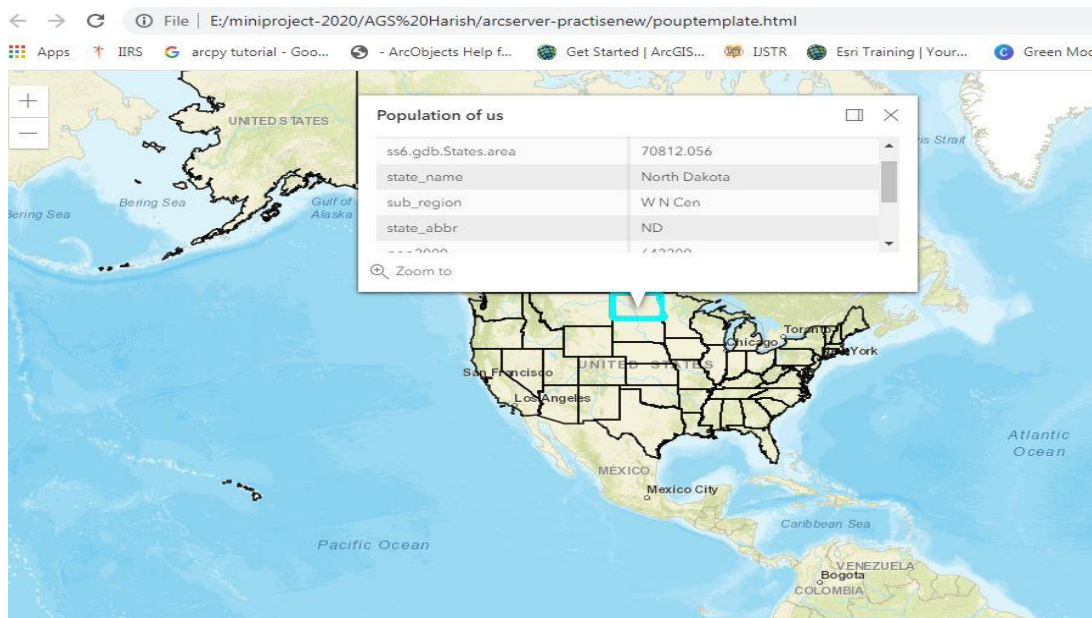


Figure 20: Popup-Template for displaying the feature information.

Using a popup template reference, facts about the selected feature are displayed in a tiny window. As the issue with the Query task is that it takes more time to learn the specifics of a feature, the popup template can help.

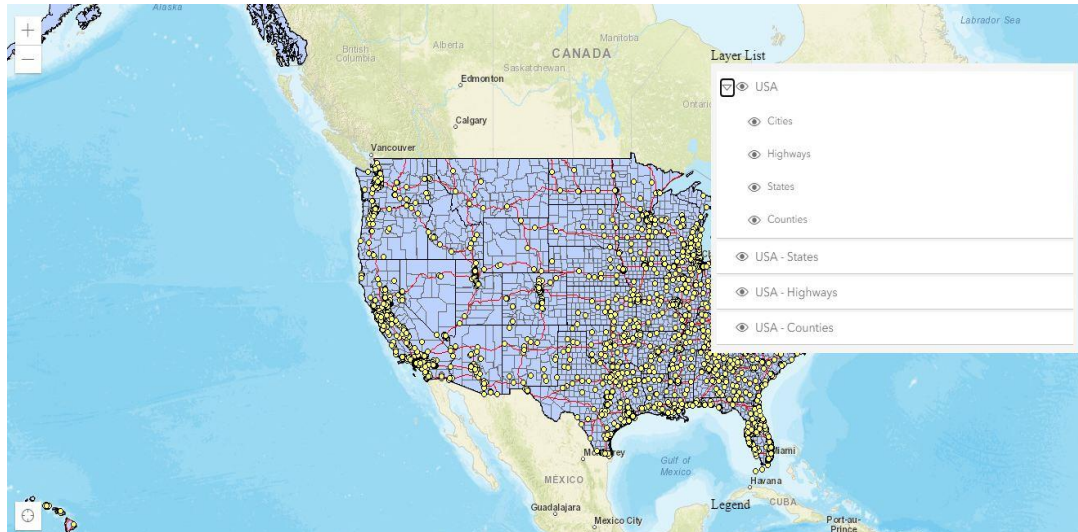


Figure 21: Layer list code for displaying the layers mentioned in source code.

Layer List code is used for appending the layers to base map mentioned in source code. The Layer-List widget provide a way to show a list of layers, and switch on/off their visibility. The List-Item API provide admission to each layer's properties, allows the developer to arrange actions related to the layer, and allows the developer to add content to the item related to the layer.

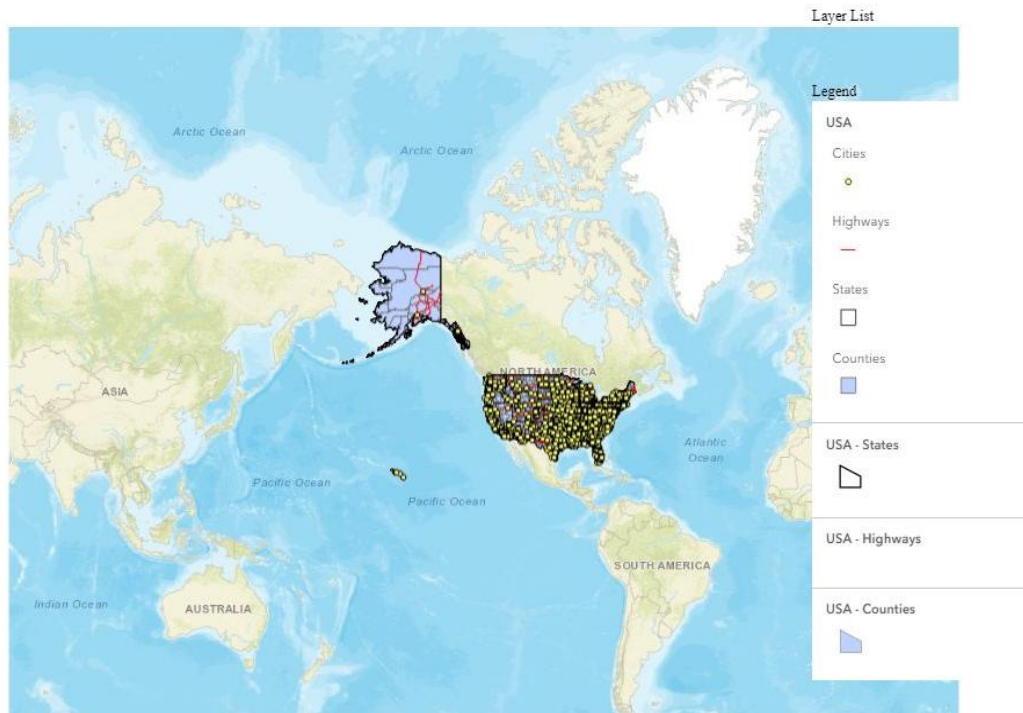


Figure 22: Legend code for displaying the features information.

Legend code is used for mentioning the information about features. The Legend widget will display labels and symbols for the layers in a map. Labels and their matching symbols depend on the values set in the renderer of the layer. The legend will only show layers and sub-layers that are detectable in the view.

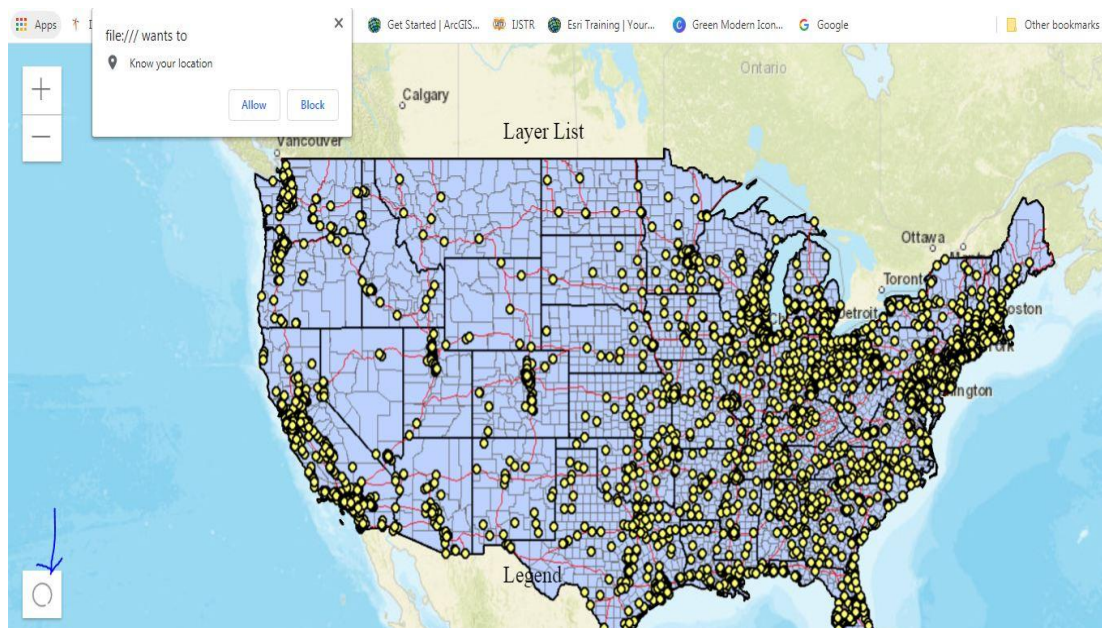


Figure 23: Locate code for navigating the base map to current location.

Locate is used for navigating to the existing location of user. It provides a simple widget that navigates the view to the user's current location. The view rotate according to the route where the tracked device is bearing towards.

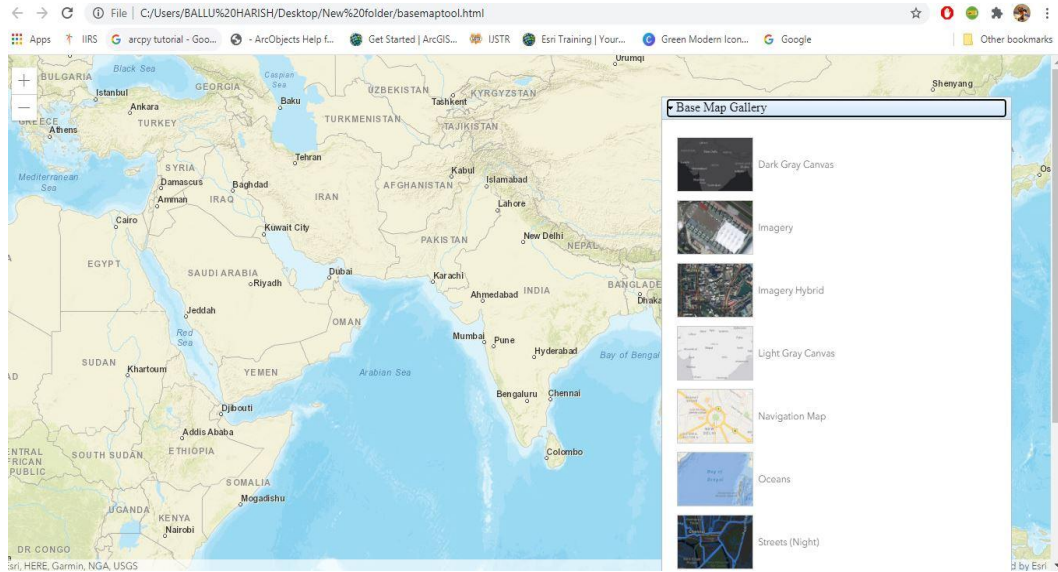


Figure 24: Base Map Gallery.

It will display all the maps at a time we can select any of them. By default Eleven different types of maps are available we can choose any of the map from them. The Base map Gallery is available under “Dojo” technique.

Whenever the form gets loaded by default the BaseMapGallery will be under false mode, by clicking on the Dojo technique we can see all the available maps.

14. Procedure for publishing the data in server: While publishing Raster data we choose Tile package, in case of vector we select publishing a service.

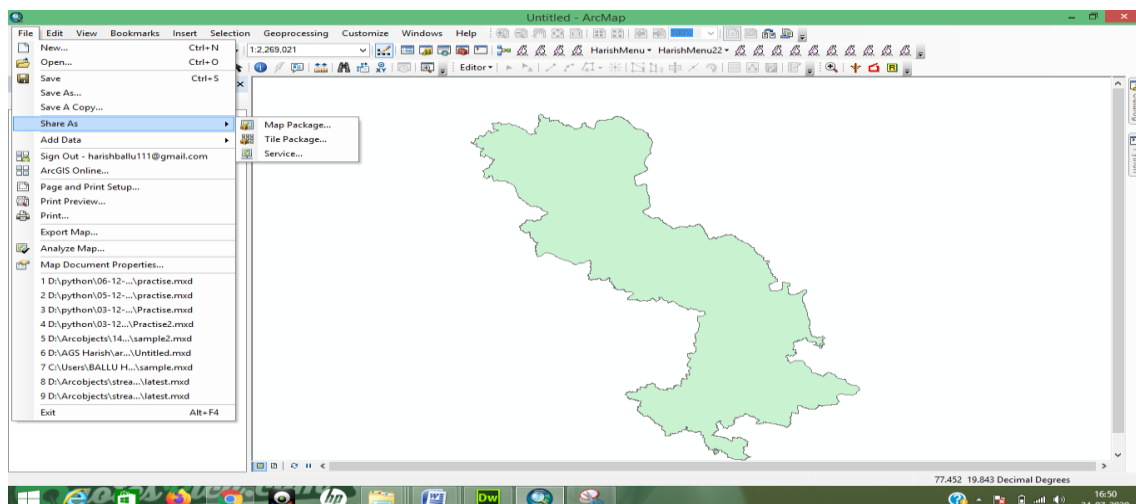


Figure 25: Publishing vector data as service.

After opening the vector data/Raster data, selecting the share as option for publishing the data.(For Raster data select Tile package, For vector data select service)

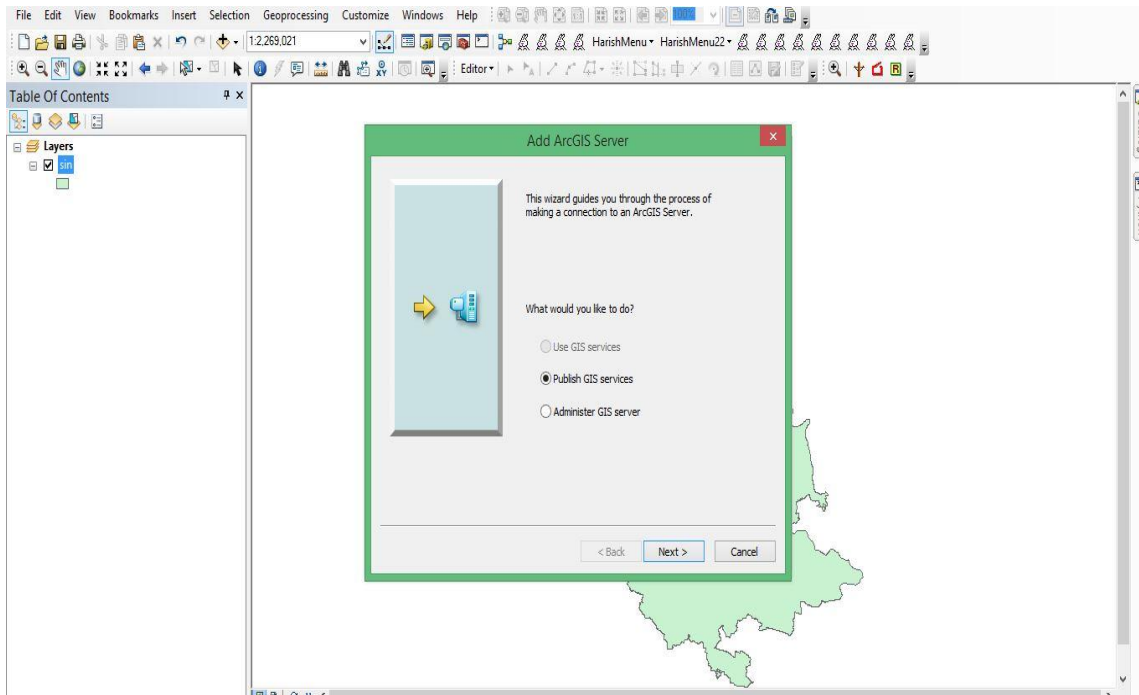


Figure 26: Selecting publishing GIS service.

By selecting the above option we are making the server ready to accept the published data.

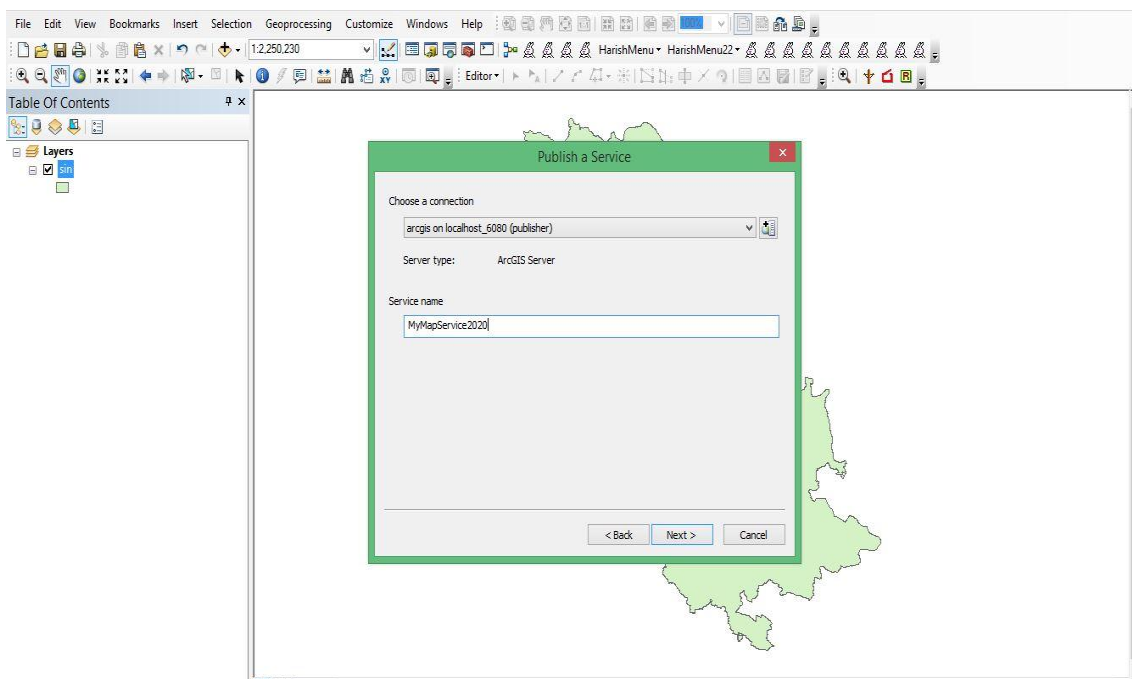


Figure 27: Selecting local/global server.

Here in this case we are selecting the server either local/global. If it is local the data gets published in our local system else the data gets published in Arc-GIS online server.

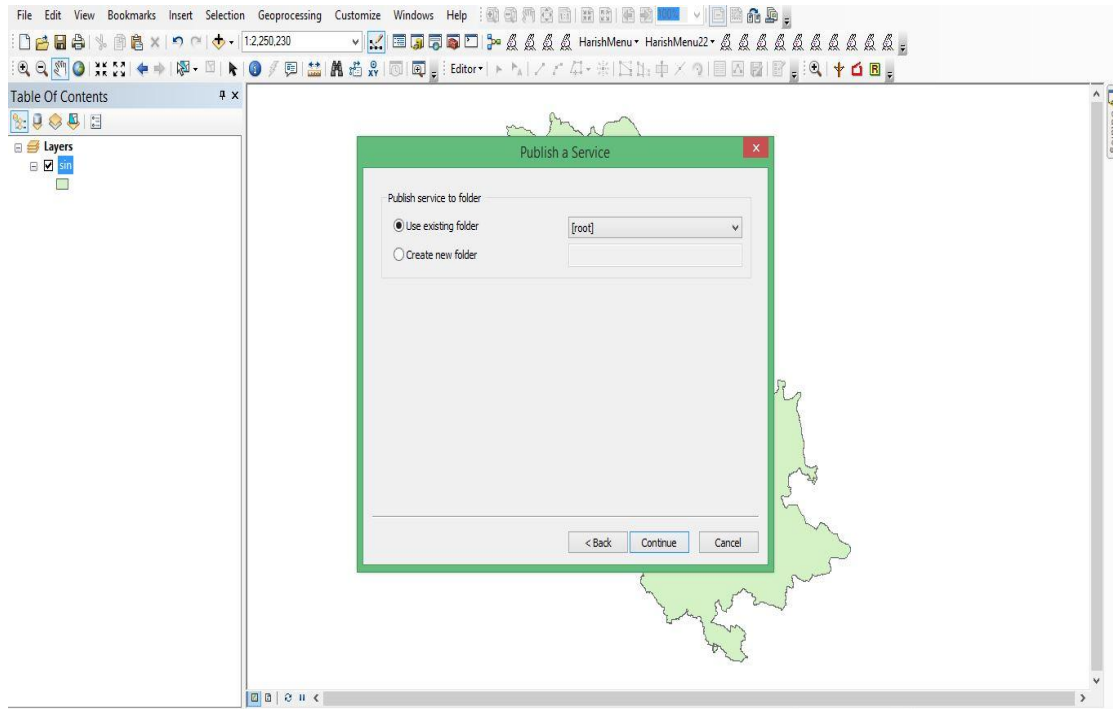


Figure 28: Choosing the folder for publishing data.

Here we are choosing the folder so that the data once getting published it will be saved in that folder. Here the data is published in root folder.

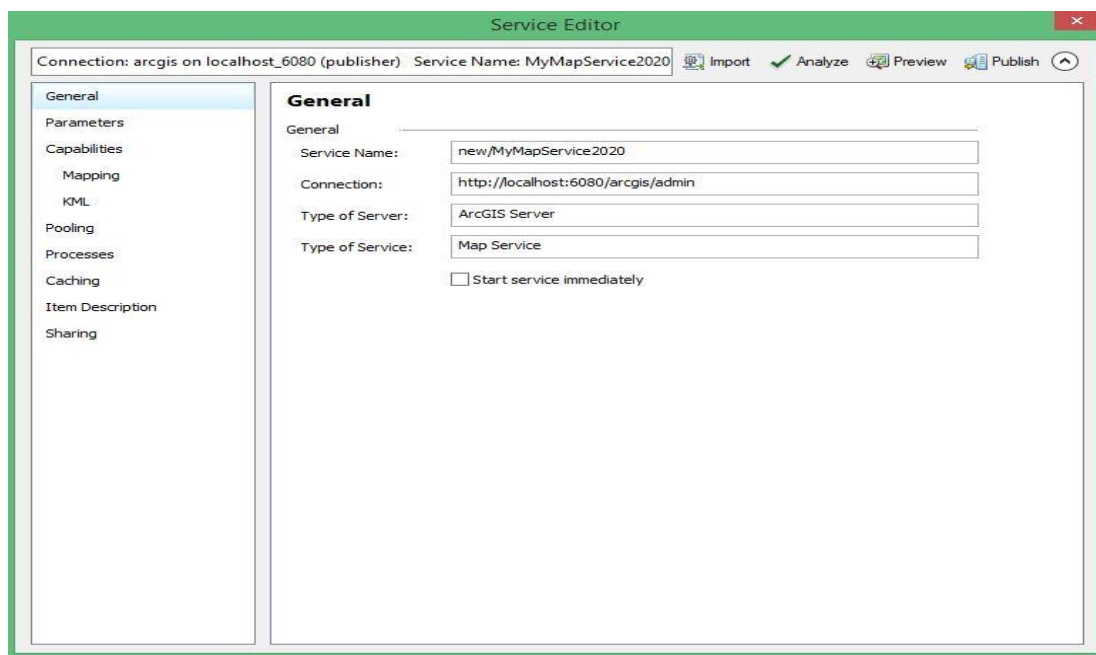


Figure29: Local server URL is known here in the connection option.

In the above figure we come across various options General tab gives basic information about the publishing data. Similarly parameters tab gives about number of records returned by user. Capabilities allow us to select various services. Cache gives information about the size of the data which is going to be published. Item description allows us to give summary and tags for the data.

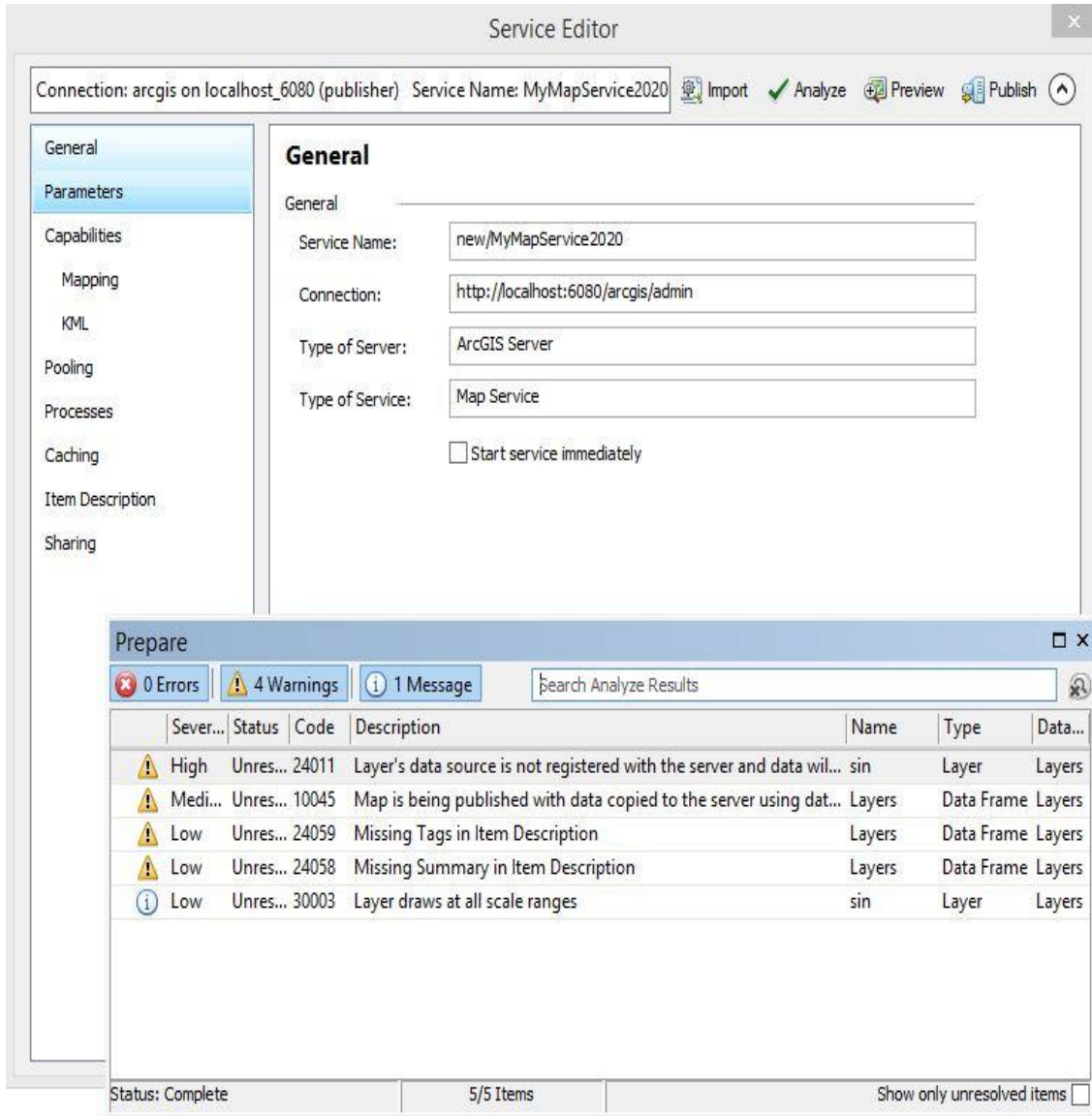


Figure 30: Validating the data which is ready to publish.

In the above figure we are validating the data which is going to be published in the server either it is following the criteria of server publishing or not.

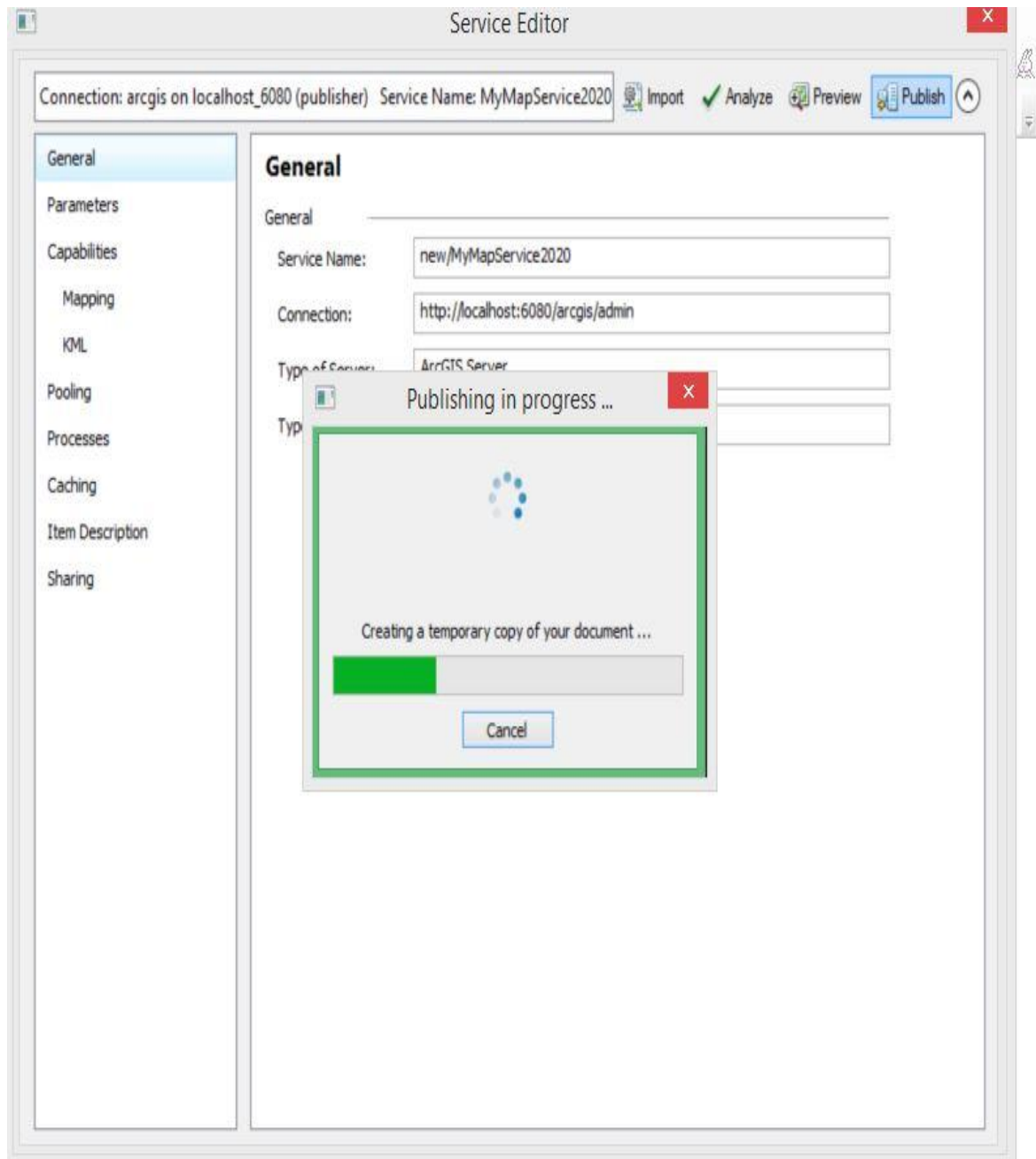


Figure 31: Successful publishing of Geo-spatial data.

Once the data is error free the data gets published in server. Once the programme becomes error free the data gets published. Regarding the errors in programme they will get displayed in red mark. By reading the errors they need to be rectified by making the corrections.

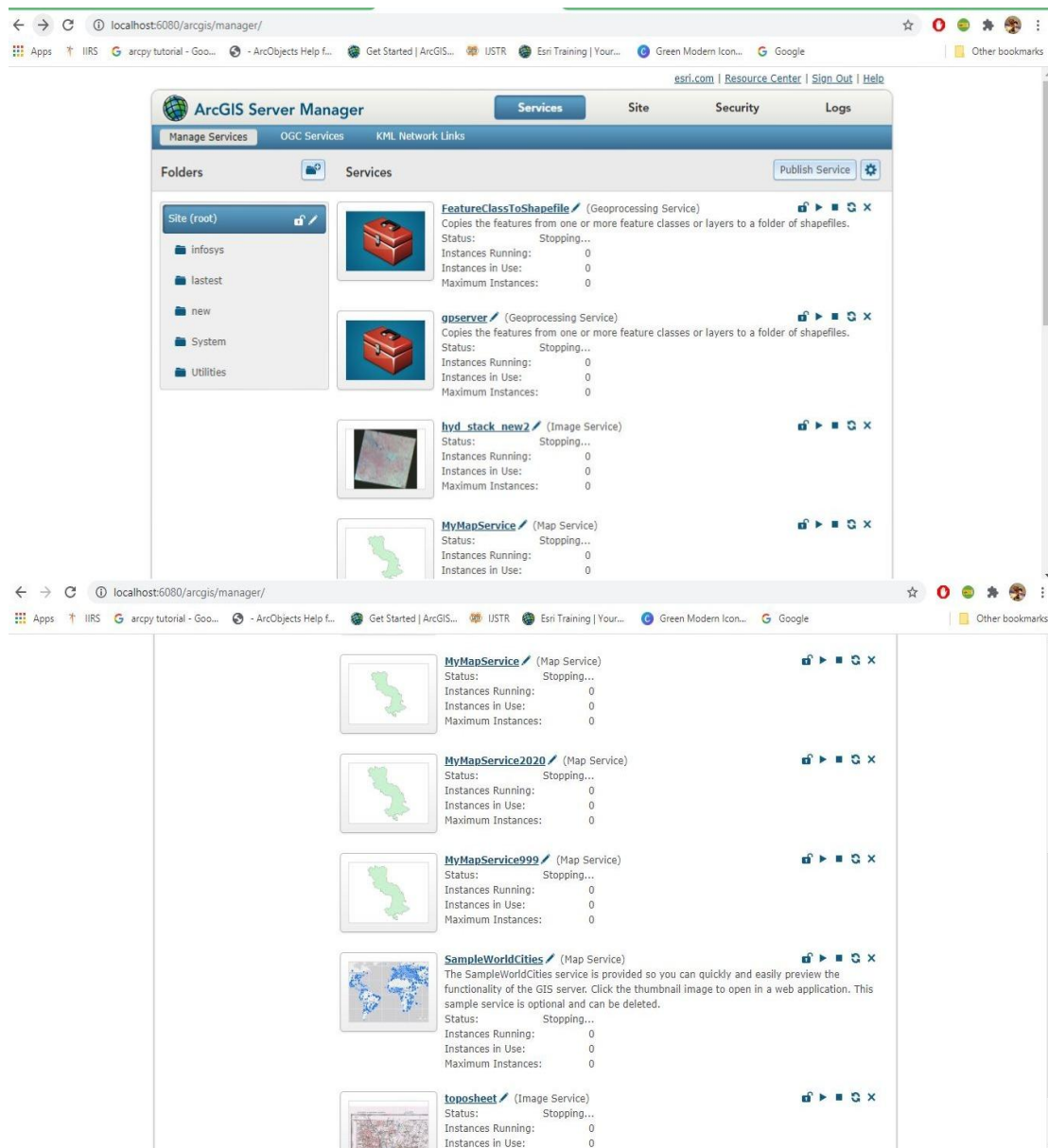


Figure 32: List of data published in server.

The above figure gives the detail information about the services which are published in server.

Arc-GIS server details:

To view the published data go through

Url: <http://localhost:6080/arcgis/manager/index.html> and enter

Username: siteadmin

Password: siteadmin

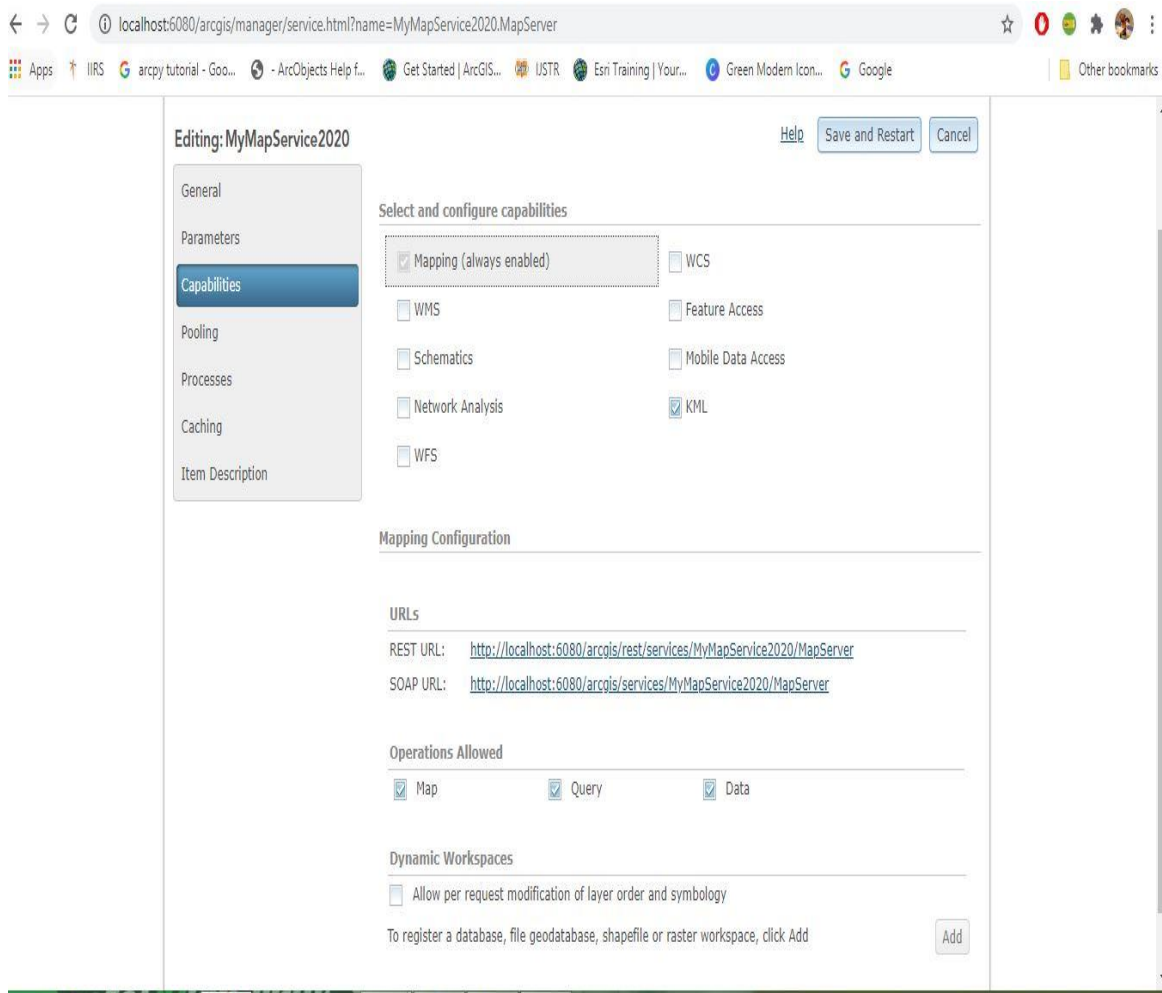


Figure 33: Url's of service can be known by above figure.

To share the data to other users we need to know the Url of the service. For that we move to capabilities tab and we can find its Url. By sharing the Url to others our data can be accessed.

The Url of above data published is

“<http://localhost:6080/arcgis/services/MyMapService2020/MapServer>“

15. Publishing of panchromatic data a part of Hyderabad region:

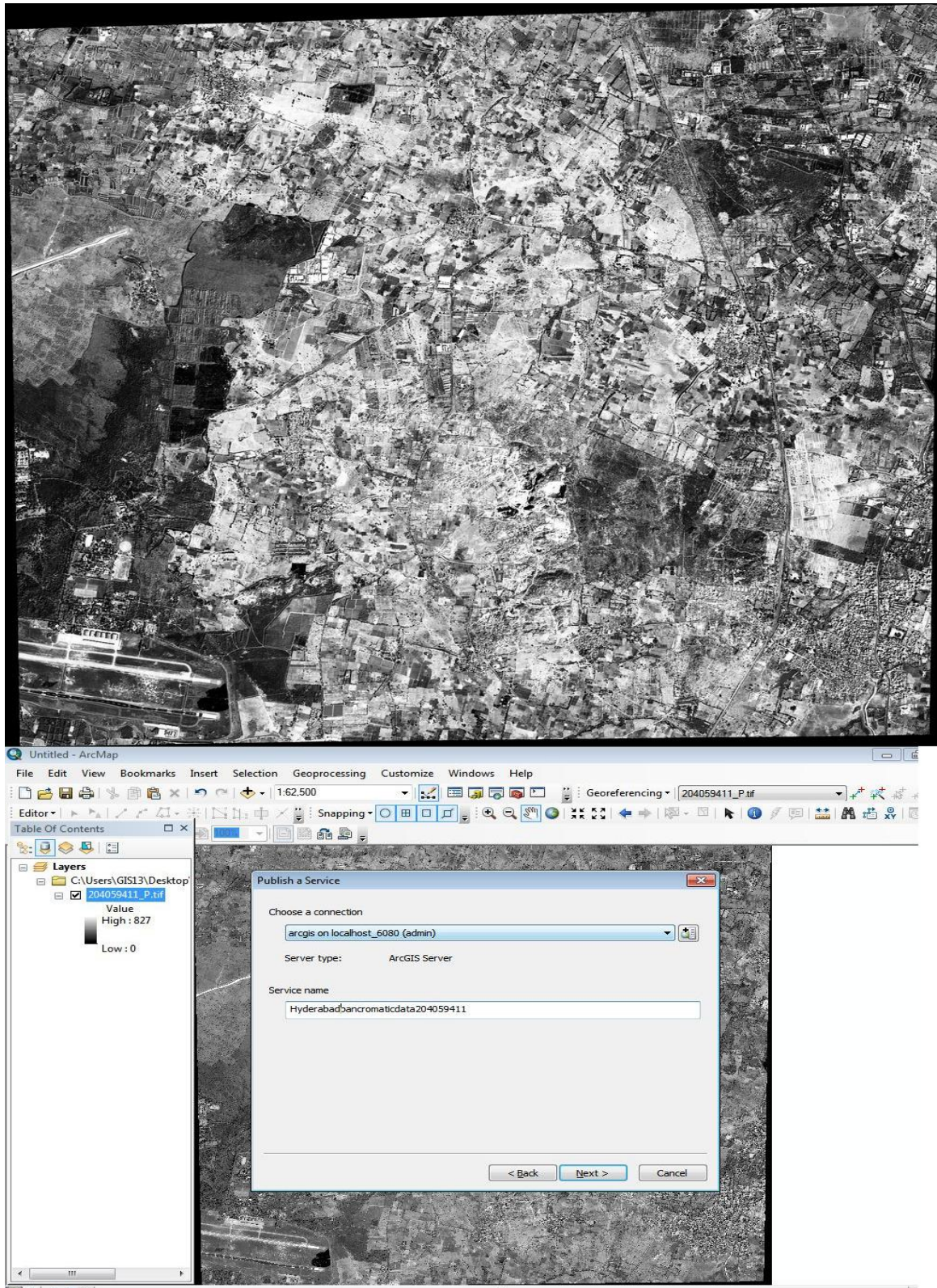


Figure 34: Part of Hyderabad regions high resolution data is selected for publishing in Arc-GIS server

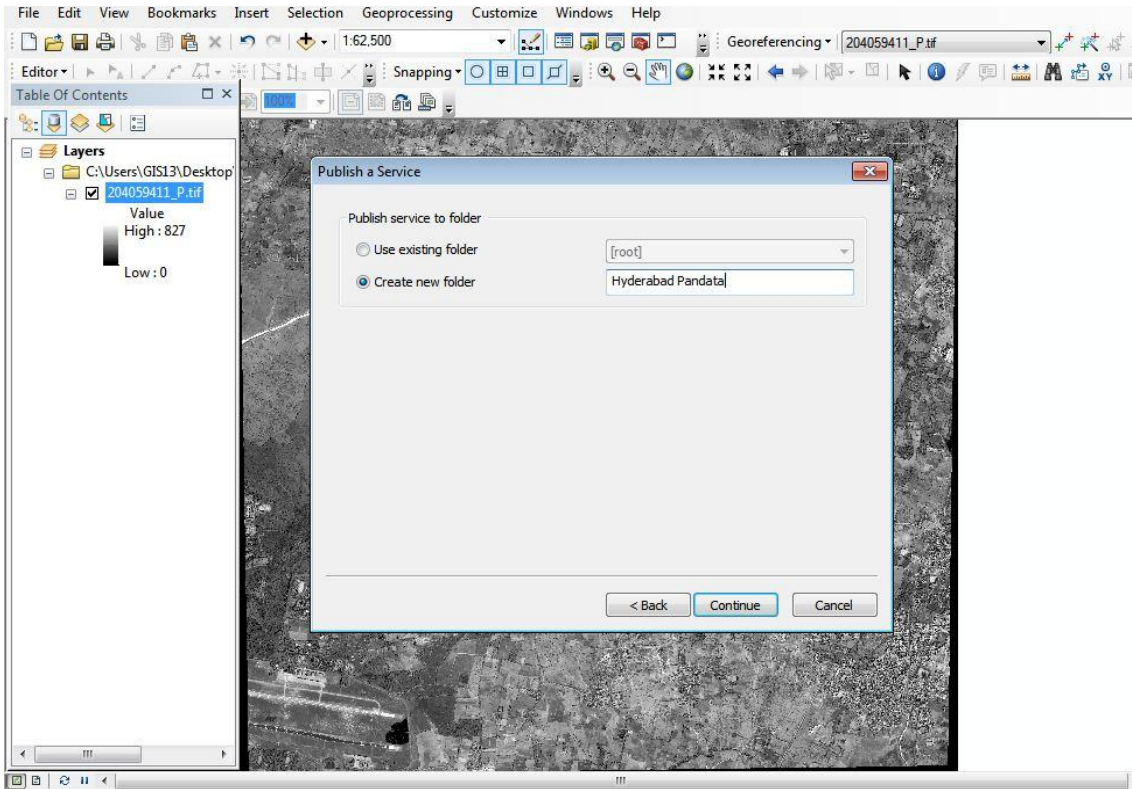


Figure 35: High resolution satellite data is going to be published under New folder named as Hyderabad Pandata in Arc-GIS server

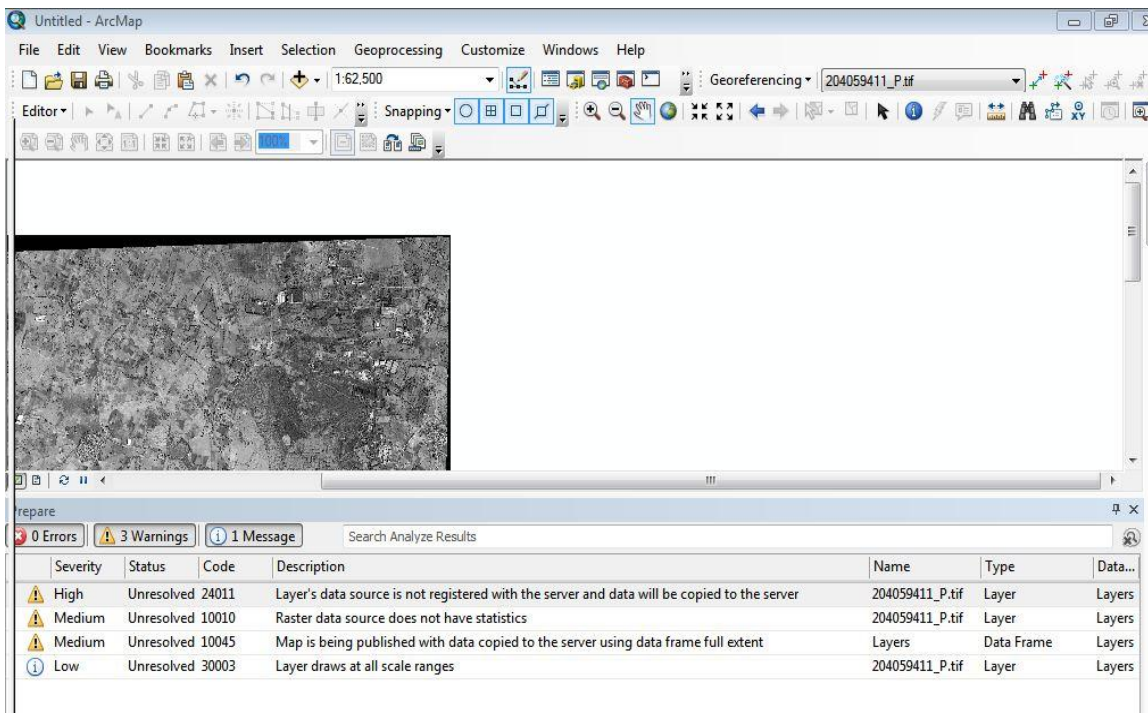


Figure 36: Checking for Errors in publishing of data

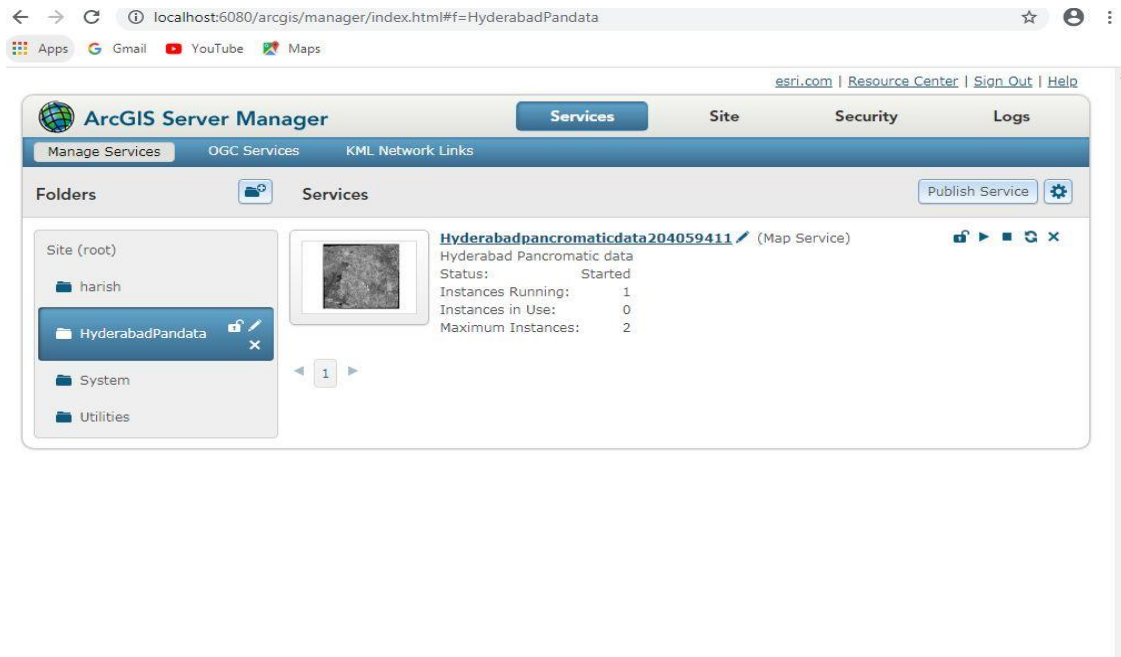


Figure 37: Publishing of High resolution satellite data is attained successfully

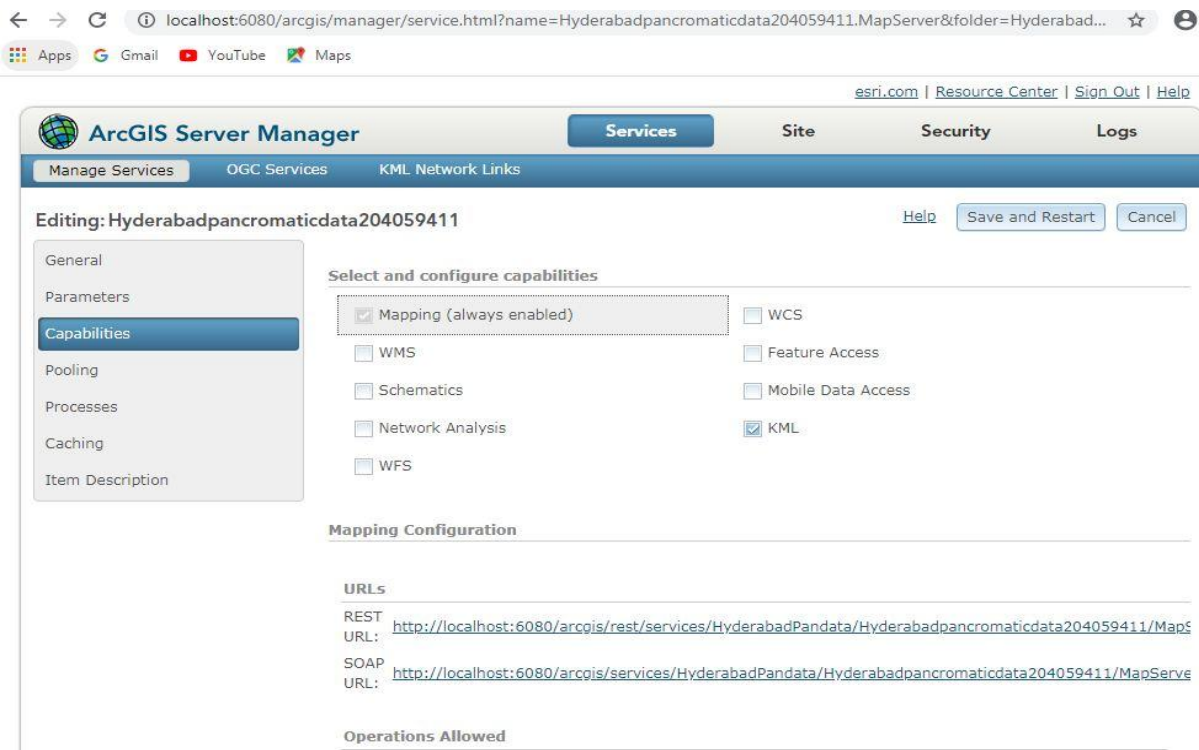


Figure 38: URL's of High resolution satellite data published in local server

The url of above data published is

<http://localhost:6080/arcgis/services/HyderabadPandata/Hyderabadpancromaticdata204059411/MapServer>

To share the data to other users we need to know the Url of the service. For that we move to capabilities tab and we can find its Url. By sharing the Url to others our data can be accessed.

16. Arc-GIS server details: To view the published data go through

url: <http://localhost:6080/arcgis/manager/index.html> and enter

Username: siteadmin

Password: siteadmin

17. Project outcomes

- A Geo Spatial database with a standard projection and a database on the Arc GIS server hub for accessing Geo Spatial data and performing customization tasks utilizing a variety of Java Script API services.
- A pop-up template is used to display specific features' details based on the zoom level.
- Validation of Geospatial data is important criteria once it is success, publishing in Arc-GIS server Environment can be attained.

18. Conclusions

- The maps that can build with **Esri JavaScript API** are changing the way that we recognize world and the way that we relate with it. Its dominant API helps keep us busy for the challenges and opportunities in the 21st century, both in interactive 2D and 3D environment.
- Just like any API, the Esri JavaScript API offers ways to make own custom applications in a diversity of ways. Every day, the Esri JavaScript API allows us to connect with our neighbourhood and our place that we're living in sole, inventive and influential ways.
- From Esri JavaScript API-3 which gave 2D webmaps to Esri JavaScript 4 which now gives the chance to build interactive 3D web-scenes has revolutionize the skill of mapping online.
- With the Esri JavaScript API, we can better notify for decision-making and represent priority from our own spatial data. We can express and claim our own potential giving our own sense of place that we're living in.

REFERENCES

- [1] Goodchild, Michael F "Crowdsourcing geographic information for disaster response: a research frontier" Pages 231-241.
- [2] Goodchild, M.F. 2007. "Citizens as sensors: the world of volunteered geography. *GeoJournal*", 69(4): 211–221
- [3] Scharl, A. and Tochtermann, K. 2007. "The geospatial web: how geobrowsers, social software, and the web 2.0 are shaping the network society".
- [4] National Research Council. 2007. "Successful response starts with a map: improving geospatial support for disaster management".

- [5] LIU Laixing, LI Deren, SHAO Zhenfeng,2008” RESEARCH ON GEOSPATIAL INFORMATION SHARING PLATFORM BASED ON ARCGIS SERVER”
- [6] Taranjot Singh Bhatia, Harpinder Singh, P.K Litoria, ,Brijendra Pateriya,2019” Web GIS Development using Portal for ArcGIS, ArcGIS Server and Web AppBuilder for ArcGIS”
- [7] Ramesh Babu R , G K Viswanathan, Tina Mary Francis,2011” Extending ArcGIS Server capabilities through customization - A technology perspective”
- [8] David J. Maguire,2008”ArcGIS: General Purpose GIS Software System”
- [9] Ibrahim Djamaluddina, Yasuhiro Mitania , Tetsukazu Yaharab, 2012 “Development of GIS Server Application for Sharing and Utilizing Biodiversity Survey Database”