1.Create SampleServlet extend the SlingAllMethod Servlet and register it using resourceType

package com.project2.core.servlets;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.servlets.SlingAllMethodsServlet;

import org.osgi.service.component.annotations.Component;

import javax.servlet.Servlet;

import javax.servlet.ServletException;

import java.io.IOException;

@Component(service = Servlet.class,

    property = {

        "sling.servlet.methods=GET",

        "sling.servlet.resourceTypes=project2/components/helloworld",

        "sling.servlet.extensions=json"

    })

public class SampleServlet extends SlingAllMethodsServlet {

    @Override

    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)

            throws ServletException, IOException {

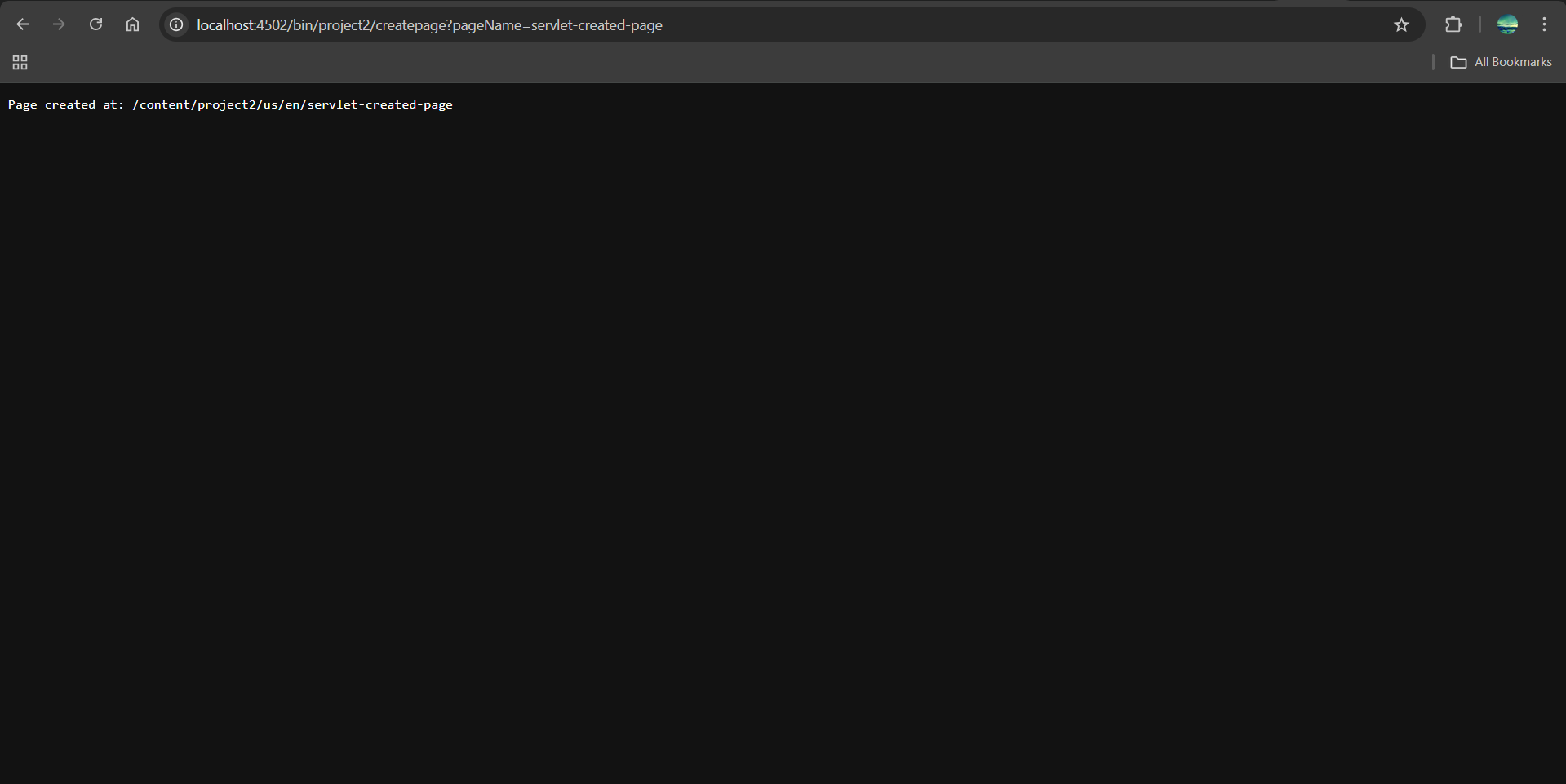
        response.setContentType("application/json");

        response.getWriter().write("{\"message\": \"Hello from SampleServlet on servlets-page!\"}");

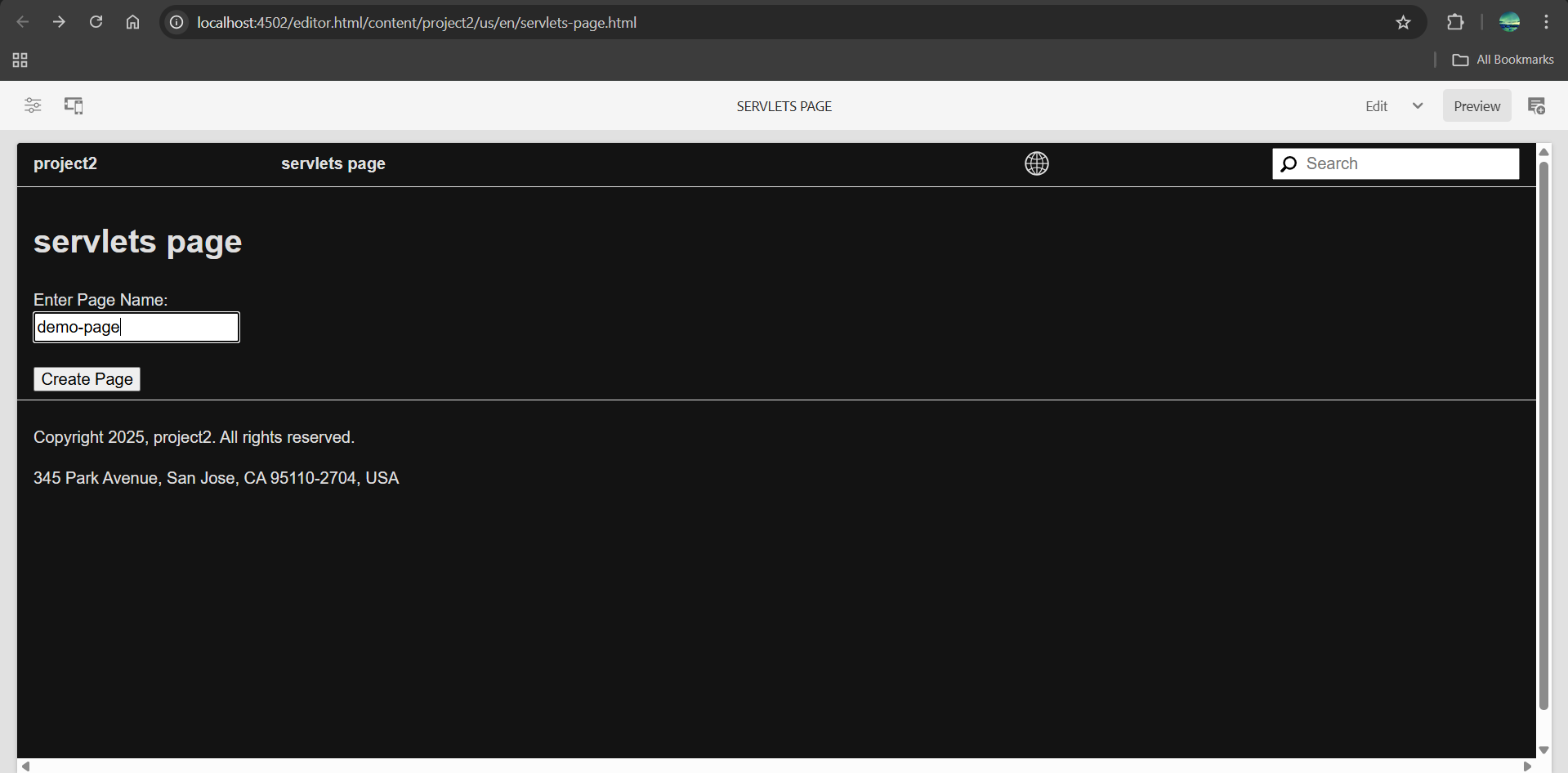
    }

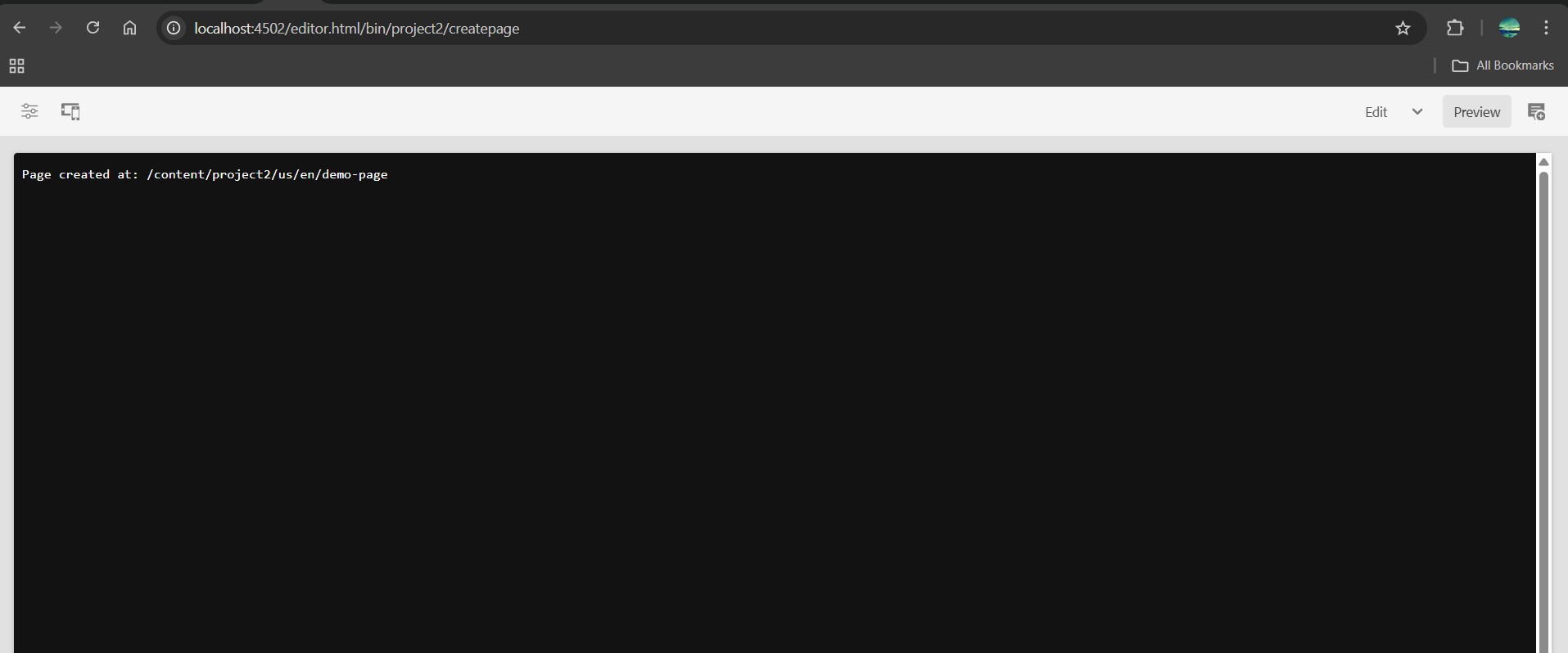
}

2.Create CreatePageServlet extend the SlingSafeMethod Servlet and register it using path



3.Take page name from user and create pages in aem using above servlet





4.Use page Manager Apis for above task.

package com.project2.core.servlets;

import com.day.cq.wcm.api.Page;

import com.day.cq.wcm.api.PageManager;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.resource.ResourceResolver;

import org.apache.sling.api.servlets.SlingSafeMethodsServlet;

import org.osgi.service.component.annotations.Component;

import javax.servlet.Servlet;

import javax.servlet.ServletException;

import java.io.IOException;

@Component(service = Servlet.class,

        property = {

                "sling.servlet.methods=GET",

                "sling.servlet.paths=/bin/project2/createpage"

        })

public class CreatePageServlet extends SlingSafeMethodsServlet {

    private static final String PARENT\_PATH = "/content/project2/us/en";

    private static final String TEMPLATE\_PATH = "/conf/project2/settings/wcm/templates/landing-page";

    @Override

    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)

            throws ServletException, IOException {

        String pageName = request.getParameter("pageName");

        if (pageName == null || pageName.isEmpty()) {

            response.getWriter().write("Page name is missing.");

            return;

        }

        try {

            // Get resource resolver from request

            ResourceResolver resolver = request.getResourceResolver();

            // Adapt to PageManager

            PageManager pageManager = resolver.adaptTo(PageManager.class);

            if (pageManager == null) {

                response.getWriter().write("Unable to adapt to PageManager.");

                return;

            }

            // Create new page

            Page createdPage = pageManager.create(PARENT\_PATH, pageName, TEMPLATE\_PATH, "Page created by servlet");

            if (createdPage != null) {

                response.getWriter().write("Page successfully created at: " + createdPage.getPath());

            } else {

                response.getWriter().write("Page creation failed.");

            }

        } catch (Exception e) {

            response.getWriter().write("Error occurred: " + e.getMessage());

        } }

}

5.Create one SearchServlet to search the content using PredicateMap to search the content from pages

SearchServlet.java

package com.project2.core.servlets;

import com.day.cq.search.Query;

import com.day.cq.search.QueryBuilder;

import com.day.cq.search.eval.PredicateGroup;

import com.day.cq.search.result.Hit;

import com.day.cq.search.result.SearchResult;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.resource.ResourceResolver;

import org.apache.sling.api.servlets.SlingSafeMethodsServlet;

import org.osgi.framework.Constants;

import org.osgi.service.component.annotations.Component;

import javax.jcr.Session;

import javax.servlet.Servlet;

import javax.servlet.ServletException;

import java.io.IOException;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

@Component(service = Servlet.class,

    property = {

        Constants.SERVICE\_DESCRIPTION + "= AEM Search Pages Servlet",

        "sling.servlet.paths=/bin/project2/searchpages",

        "sling.servlet.methods=GET"

    })

public class SearchServlet extends SlingSafeMethodsServlet {

    @Override

    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)

            throws ServletException, IOException {

        String keyword = request.getParameter("q");

        if (keyword == null || keyword.trim().isEmpty()) {

            response.getWriter().write("Please provide query string like ?q=article");

            return; }

        ResourceResolver resolver = request.getResourceResolver();

        Session session = resolver.adaptTo(Session.class);

        QueryBuilder queryBuilder = resolver.adaptTo(QueryBuilder.class);

        if (session == null || queryBuilder == null) {

            response.getWriter().write("QueryBuilder or session is null");

            return;

        }

        // Define query predicates

        Map<String, String> predicates = new HashMap<>();

        predicates.put("path", "/content/project2/us/en/news");

        predicates.put("type", "cq:Page");

        predicates.put("fulltext", keyword);

        predicates.put("p.limit", "-1");

        PredicateGroup predicateGroup = PredicateGroup.create(predicates);

        Query query = queryBuilder.createQuery(predicateGroup, session);

        SearchResult result = query.getResult();

        List<Hit> hits = result.getHits();

        response.setContentType("application/json");

        StringBuilder json = new StringBuilder("{ \"results\": [");

        for (int i = 0; i < hits.size(); i++) {

            String path = hits.get(i).getPath();

            System.out.println("Found path: " + path);

            json.append("\"").append(path).append("\"");

            if (i < hits.size() - 1) json.append(", ");

        }

        json.append("]}");

        response.getWriter().write(json.toString());

    }

}