# NCI Web Application Development

*Group Project*

|  |  |
| --- | --- |
| **Group Name** | |
| Calendar Map Team | |
| **Student Names** | **GitHub Accounts** |
| MARK GASKIN - 16111931 | Markog222 |
| STEFAN DWORSCHAK - 15037835 | stefdworschak |
| GRAHAM FARRELL - 15044726 | GrahamFarrell1984 |
| **Link to your final Workspace in Codeanywhere (just one)** | |
| <https://calendar_nodeengine-stefdworschak632650.codeanyapp.com/> | |
| **Node.js modules used in the project (if applicable)** | |
| @google/maps, body-parser, ejs, express, express-session, fs, js2xmlparser,  lbxmljs, libxslt, simple-encryptor, xml2js | |
| **GitHub Repository** | |
| <https://github.com/stefdworschak/web_development_nci> | |

## **Purpose of Document**

This document is to provide a brief report outlining each of the technologies used in the project, where they were used and what were the advantages of using the chosen technology over another.

## 

## **Intended Audience**

Mikhail Timofeev

## **Project Report**

**Executive summary:**

Based on the initial review of this project the team took the line of developing a mapping tool within a website. The team examined the functionality of the google API for maps and GEOCODE which was incorporated into the application. During early discussions it was decided to change focus of the application to a Calendar mapping and notation tool.

The first page allows the user to complete the following tasks:

* Log in
* Create a new user

The second [main] page allows the user to complete the following tasks:

* View any current calendar entries on the map
* Add new events
* View Data List for previous entries in the table format
* Delete entries from the data list table format
* Share your calendar entries with friends by entering their email address
* Log Out
* Change Password
* View shared calendars
* Search for entries
* Hover over map markers for more information

The final application provides the user, be it for personal or work use, with the ability to log upcoming events so they can be tracked and logged within the main page map. They can also share their entries with friends or colleagues to show their current availability or current location.

**Technologies used and their advantages:**

Google Maps API

The well-known service provided by google and used by billions was a key tool utilised within our project. The implementation of this tool was difficult at first to work into the project but with the multitude of available tutorials online the team was able to incorporate the mapping tool.

Starting off, to use mapping service from google requires you to create a google account. Once signed in you can follow some easy steps to generate the map API which was built into our index.ejs code. Built into the background of the google map API is the ability to share addresses and add markers for newly entered locations. In comparison to other mapping API’s, such as MapQuest or Bing Maps, the Google Maps API had much more online resources and provided more follow-on functionality.

Google Geocode API

The GEOCODE API used in this project was implemented in the same way as the Maps API. The GEOCODE allowed the team to extract the exact location of any new entries and log this within the XML. This data was then saved in our data list table. Again, like the Maps API the GEOCODE API provided much more follow on functionality then other GEOCODE services like MapQuest, Bing, Yahoo and many others.

Bootstrap

Bootstrap was an easy way for the group to provide a sleek and well framed website with using only a handle full of extra lines of code and div’s. The team utilised the Bootstrap Glyphicon and Modal within both the log in page and the main page. For each of the log in, add new event, data list, share calendar and multiple section button we added a Glyphicon that suited the task being selected. The modal format allowed for a smooth output of the functions or data within a confined area. Since Bootstrap is an open source search provided and is easily implemented within any HTML, CSS or JS the team did not look anywhere else for a similar output for the project.

XML

XML (Extensible Markup Language) was used within the project to allow our application to take the inputs from the user from the Enter New Event section and then store that information in an XML document which the user can then view in the table in the View Data List section. It stores an id, a date, a time, a description of the event, the full name of the user, the location where the event is taking place and also the latitudinal and longitudinal coordinates of that location.

JSON

Our group used JSON (JavaScript Object Notation) in the project to be able to send information to the server and get information back from the server so that our application can set the markers for the appointments on the map. Like the XML document, it uses an id, a date, a time, a description of the event, the full name of the user, the location where the event is taking place and also the latitudinal and longitudinal coordinates of that location.

XSD

XSD (XML Schema Definition) was used alongside our XML to define the structure, elements and attributes in our XML document and to also ensure that our XML document is valid.

XSL

Our group also used XSL (Extensible Stylesheet Language) to transform our XML to HTML before displaying it in the browser.

NODE.JS DEPENDENCIES

@google/maps: used for geocoding appointments’ location

Body-parser: middleware used for to read POST data

EJS: templating engine used in combination with express

Express: used for creating the overall server structure and the routing

Express-Session: used for the login feature; storing user details in a cookie based session

FS: used for reading/writing to the JSON/XML/XSD/XSL files used as storage for the project

Js2xmlparser: used to convert JSON to XML.

Libxmljs: used to validate the XML file against an XSD Schema

Libxslt: used to convert the XML to html using the XSL file

Simple-encryptor: used to encrypt and decrypt the user password

Xml2js: used to convert XML to JSON