MAP BASED APPLICATION ON AMERICAN COLONIZATION

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Salil Aggarwal
Spring 2015
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Salil Aggarwal:

Map Based Application on American Colonization

Carl Eckberg, Chair
Department of Computer Science

William Root
Department of Computer Science

Carmelo Interlando
Department of Mathematics and Statistics

April 20, 2015
Approval Date
ABSTRACT OF THE THESIS

Map Based Application on American Colonization
by
Salil Aggarwal
Master of Science in Computer Science
San Diego State University, 2015

The aim of this thesis is to investigate American Colonization, that is the discovery and colonizing of America. American colonization often refers to American westward expansion, acquisition of the Philippines, etc. By using the word “Colonization” that ambiguity is hopefully avoided.

Colonization of Americas began in 1492 by European when Spanish voyages headed by Christopher Columbus (an Italian Sailor) sailed west to find a new trade route to the East but unexpectedly found the Americas.

The primary goal of this thesis is to explore Google map facilities to build a map based application showing the colonial period on an interactive timeline, including maps to show Norse exploration and Columbus four voyages to the west in search of new trade routes when he unknowingly found the Americas. To show on maps the original thirteen colonies that were setup in the United States and are divided into three colonial regions – New England, Middle Colonies and Southern colonies.

The application can be accessed using any standard web browser and informs the user about the colonization and colonial period of the United States. Maps are implemented to make the application more responsive to the user.

Technologies used are: JavaScript, JQuery, HTML, CSS and Google maps APIs.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT ........................................................................................................ iv</td>
</tr>
<tr>
<td>LIST OF FIGURES .............................................................................................. vii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS ......................................................................................... viii</td>
</tr>
<tr>
<td>CHAPTER</td>
</tr>
<tr>
<td>1 INTRODUCTION ................................................................................................. 1</td>
</tr>
<tr>
<td>2 TECHNOLOGIES USED IN APPLICATION ............................................................ 3</td>
</tr>
<tr>
<td>2.1 HTML5 (Hyper Text Markup Language, Version 5) ...................................... 3</td>
</tr>
<tr>
<td>2.2 CSS3 (Cascading Style Sheets Version: 3) .................................................... 4</td>
</tr>
<tr>
<td>2.3 JavaScript ...................................................................................................... 4</td>
</tr>
<tr>
<td>2.4 JQuery ........................................................................................................... 5</td>
</tr>
<tr>
<td>2.5 Google Maps API V3 .................................................................................... 5</td>
</tr>
<tr>
<td>2.6 To Run The Application ............................................................................... 5</td>
</tr>
<tr>
<td>2.7 Supported Browsers ..................................................................................... 6</td>
</tr>
<tr>
<td>3 RESEARCH AND DESIGN .................................................................................. 7</td>
</tr>
<tr>
<td>4 AMERICAN COLONIAL ERAS ........................................................................... 10</td>
</tr>
<tr>
<td>4.1 Before Columbus - Native Americans ......................................................... 11</td>
</tr>
<tr>
<td>4.2 Christopher Columbus (1492- 1600) ............................................................ 11</td>
</tr>
<tr>
<td>4.3 Virginia Dare (First Child born in the New World) ...................................... 11</td>
</tr>
<tr>
<td>4.4 Salem witch trials ......................................................................................... 11</td>
</tr>
<tr>
<td>4.5 Anne Bradstreet ............................................................................................ 12</td>
</tr>
<tr>
<td>4.6 Seventeenth Century ..................................................................................... 12</td>
</tr>
<tr>
<td>4.7 Eighteenth Century ..................................................................................... 12</td>
</tr>
<tr>
<td>5 LOOK AND FEEL OF THE ENTIRE APPLICATION ............................................ 13</td>
</tr>
<tr>
<td>5.1 Home Page ................................................................................................... 13</td>
</tr>
<tr>
<td>5.2 Maps ............................................................................................................ 14</td>
</tr>
</tbody>
</table>
5.3 Timeline .................................................................................................................16
5.4 Gallery ..................................................................................................................17

6 MAP IMPLEMENTATION .........................................................................................19
   6.1. Columbus Voyages Map .......................................................................................19
   6.2. Thirteen Colonies and Three Colonial Regions (New England, Middle Colonies, Southern Colonies) Maps ..............................................................................27
   6.3. Adding Custom Paths to the Map ..........................................................................36

7 SUMMARY, OBSTACLES, FUTURE WORK, AND TESTING ..............................47
REFERENCES .............................................................................................................49
LIST OF FIGURES

Page

Figure 3.1. Custom Interface to add custom paths on map.........................................................9
Figure 4.1. American colonization eras. ...............................................................10
Figure 5.1. Home page of the application...............................................................13
Figure 5.2. Map 1: Columbus Voyages. Map showing Columbus voyages..............14
Figure 5.3. Map 2: Thirteen original colonies. Map showing thirteen colonies and three colonial regions.................................................................15
Figure 5.4. Map 3: Norse voyages.....................................................................................15
Figure 5.5. Map 4: Custom voyage interface........................................................................16
Figure 5.6. Timeline showing different American Colonial Eras..............................17
Figure 5.7. Application’s Gallery, includes books and videos...........................................17
Figure 6.1. Columbus Voyages Map.................................................................................19
Figure 6.2. Thirteen colonies and three colonial regions.................................................27
Figure 6.3. Adding custom voyages to the map...............................................................36
ACKNOWLEDGEMENTS

I would like to express my special thanks and gratitude to my committee chair Professor Carl Eckberg for advising and giving suggestions throughout my thesis. Without his guidance and help this dissertation would not have been possible.

I would like to thank Professor William Root for being part of the thesis committee.

Also, I would like to thank Professor Carmelo Interlando for being part of the thesis committee.
CHAPTER 1

INTRODUCTION

Almost everyone in America learns about the discovery of America and the early years of the United States.

This thesis presents this information through dynamic maps with multimedia accompaniment, focusing on composing visual renderings of these topics. The goal was to explore Google maps facilities, as well as multimedia facilities to show the story of early history of America in the most visually compelling way possible.

Below are the Google map facilities explored during building of the application.

- Google map markers [1]
  Google map marker adds a marker to the map and can be customized to different display icon options; one can customized it to custom labels as well by adding “.png” file to marker constructor google.maps.Markers. Marker constructor takes a single Marker object literal specifying the initial properties of the marker.

- Google map polylines [2]
  Google map polylines are used to add path lines to the map. Polylines are formed as a set of latitude and longitude pairs. To draw a polyline between two points, one needs to specify starting point latitude/longitude and end point latitude/longitude. Polylines can be customized to different colors and different symbol path, which is an icon that runs over the polyline from start point to end point.

- Google map Polygons [3]
  Google map Polygons are used to create polygonal figures on the map. A simple polygon is created by defining latitude and longitude coordinates of polygon’s path. In our application American Colonial regions are presented on map using Google map polygons. Google map polygons are customized to add various features like Info windows popups, etc.

  To enhance the Geographical Information System world, interface is developed to let user display custom paths on the map. A user can present the information like flight path, train circuits, etc. on map by providing initial and destination longitudinal latitudinal points
and selecting the specific path icon. The tool can be extended to build applications where visual presentations (like displaying complex path on maps) are required.

The next chapter includes all the technologies used in the application and chapter three describes the research and design of the application explaining research which was done to gather information and design the application to represent information in the most useful way.

Chapter four shows selected history snippets to give an idea to the reader about the American Colonization. Colonization eras were presented using an interactive timeline in the application.

Chapter five describes the look and feel of the entire application by demonstrating home page, timeline, maps, gallery and other parts of the application.

Chapter six shows the actual implementation of the map in the application using Google maps APIs and also provides detailed information about the enhancement interface added to create custom paths.

Last chapter covers the summary, obstacles, and testing and future enhancement of the application.

Let’s get started with each one of them.
CHAPTER 2

TECHNOLOGIES USED IN APPLICATION

This chapter provides all the detailed information about the software’s used for creating this application. The application includes the following technologies:

- HTML5
- CSS3
- JQuery
- JavaScript
- Google Maps JavaScript API v3
- Webstorm (IDE)

All the technologies/ software’s mentioned above are open source and one can use them for free to build an application.

2.1 HTML5 (HYPER TEXT MARKUP LANGUAGE, VERSION 5)

HTML5 is a markup language used for building content for World Wide Web. It supports all the standard browsers and also supports the mobile browsers [4]. It supports all the multimedia (image, video, audio) to embed in the web pages while keeping it easy to read by humans and at the same time understood by computers.

Comparing to its predecessors, HTML5 adds new features like including new tags (<canvas>, <video>, <audio>), introducing elements and new APIs that specifies its use with JavaScript.

To check the compatibility with old browsers, HTML5 is designed to handle errors by simply ignoring its new constructs.

In all, it is an improved and powerful markup language used to build web content and present the content for World Wide Web supporting all the standard web browsers.
2.2 CSS3 (Cascading Style Sheets Version: 3)

CSS is a styling language used for writing styling rules to define the look and feel of the document. It is used to create visually engaging pages and interfaces for web applications [4].

We can add styling scripts to change colors, layouts and fonts of the content, which improves accessibility and makes content more presentable to the user.

A single CSS can be shared among multiple HTML pages to reduce complexity of the web application and make it more flexible to change or reconfiguration.

The World Wide Web Consortium maintains specifications of CSS. We can embed CSS inside any HTML file or can make a separate CSS file with extension .css containing style rules.

2.3 JavaScript

JavaScript is a programming language of HTML. By using JavaScript we can make webpages do what we want them to do i.e. to define the behavior of web pages.

JavaScript is a scripting language which improves the performance of your web application by running at client side and can be used to create alerts, pop-up messages, defining functionality to buttons, navigation between web pages and much more in your web application or website [5].

It is a prototype-based scripting language that supports functional and object oriented programming styles and can also be used with desktop widgets or powering PDFs.

It has standard libraries very similar to the Java programming language, but its syntax is actually derived from C.

It is not restricted to any browser but should be turned on to use by any browser.

Almost all web applications rely on JavaScript to provide functionality like triggers, button-clicks, and on-going events to the webpages to make the application more interactive.
2.4 JQUERY

JQuery makes use of JavaScript simpler; it is basically a JavaScript library.
It is open source software which is designed and developed to reduce complexity in
JavaScript code by providing shared libraries to the developers to implement functionalities
like Ajax interaction, animation or to handle events.
It is the most popular library for JavaScript today and provides a consistent interface
to behave the same across different browsers by handling cross-browser inconsistencies.

2.5 GOOGLE MAPS API V3

It is an application-programming interface developed by Google to embed Google
maps to our web application and customize it according to use [6]. We can customize Google
maps by adding markers to show marks/points of whatever information we want to show on
maps. We can change the type of map to Roadmap, Satellite map, Hybrid map or Terrain
map.

We can modify the map by changing its base color, colors of roads/terrains and can
zoom it to any value. We also can set the point of reference by providing latitude and
longitude of the point.

It is a free source API provided by Google. All the markers, path, information tabs,
regions used in the application are designed using this Google map API.

2.6 TO RUN THE APPLICATION

- Download Webstorm IDE or any other IDE to run the application.
- Add the entire HTML files including all CSS, JQuery, video and image files to
  the IDE.
- Application will run using localhost on web browser.
- Navigate between different web pages using the application Menu bar at the top.
- Make sure to connect to Internet to see all the functionality of application like
  maps and videos.
2.7 SUPPORTED BROWSERS

Below are the supported web browsers for the application, make sure to install or update the adobe flash player.

- IE10 or above
- Firefox 24 or above
- Chrome 30 or above
- Safari 6 or above
CHAPTER 3

RESEARCH AND DESIGN

Extensive research was done to gather information on American colonization history using the Internet, consulting archeological professors and some history major friends.

Out of all the information gathered, selective information is presented using an interactive timeline. Timeline is designed to categorize different colonization eras starting from very earlier stage of American colonization i.e. era of Native Americans and then showing the Columbus voyages and other events happened during the colonization like birth of first Christian baby (Virginia Dare) on new world, Salem witch trails, etc.

Information like Columbus, Norse voyages which include the movement of people from other part of the world to explore new places was a major challenge to show on the application as information like this should be presented in more graphical and interactive way to explain and let user know about their movements as informative and as clear as possible, so after detailed research, Google maps were selected to present map information.

Google maps were selected because of its robust functionality and extensive customized object libraries which let us customize maps and information on maps using several types of map overlays like markers, polyline, polygon, Info Windows and many more custom overlays.

To display the Norse and Columbus voyages, Google polylines are used, initial and destination longitude latitude points are provided to display the path between two points, to show the complete voyage and array data structure is implemented which includes all the initial and destination points, path color, customized icons, display text (hover text), etc. To show initial/destination points, Google markers are used; marker icons are customized to give ancient voyage look.

To show original thirteen colonies that were setup in the United States, Google markers are used to show the marks of all thirteen colonies at their longitude and latitude
points. Customized icons (Flag icon) are used to show marking points of colonies on the map to make it more attractive.

To show original three colonial regions, Google map polygons are used. Polygons are created by defining LatLong coordinates for the polygon’s path. An array structure containing detailed information about polygon’s coordinates, stroke color, fill color is defined and used to display three colonial regions (New England, Middle Colonies and Southern colonies). An info window is added to all three colonial regions i.e. a user can click on colonial region to get more detailed information about that particular region. An image of United States map highlighting the colonial region is added in info window to provide an idea about the colonial region location.

An extensive research was done to find books and video on American colonization and selected books and videos are added to the application’s gallery.

To enhance the Geographical Information System i.e. to make the system more advanced and useful for the users, an interface is developed where a user can display information like flight path, train circuits, etc. on map by providing initial and destination longitudinal latitudinal points and can customize the informational path (like icons, path color, etc.) according to the needs.

The interface is built using Google maps API as a base platform, which can be further extended by anyone to include in their application to present information that requires map paths to make an application more visually compelling for example an application on world’s airports or country’s train circuit or country’s highways.

The interface looks like Figure 3.1.

Below are the inputs required by the interface to create a path on map.

- Start point latitude
- Start point longitude
- End point latitude
- End point longitude
- Select the type of path: Ship/ Flight/ Train (can be extended)

Interface can be extended to add series of path by using an array structure to store longitude and latitude of different points.
The next chapter shows selected history snippets to give an idea to the reader about the thesis, explaining different American eras of colonization.
CHAPTER 4

AMERICAN COLONIAL ERAS

Figure 4.1. American colonization eras. Sources: [7-15].
Colonization eras are presented in the application using an interactive timeline, below are the selected colonization eras of American colonization history.

**4.1 BEFORE COLUMBUS - NATIVE AMERICANS**

Native Americans are people with various distinct, ethnic groups and bands who have been living within the boundaries of the United States since the beginning of the new world and before Columbus explored America [16]. Newly announced United States go up against British for allegiance of Native American nations during the American revolution. The majority of Native Americans supported the British based on the trade relationship and hoping to get colonial expansion of Native American land.

**4.2 CHRISTOPHER COLUMBUS (1492-1600)**

Christopher Columbus (an Italian explorer and navigator) was a citizen of the Republic of Genoa. Sponsored by Catholic Monarchs of Spain, he sailed four voyages across the Atlantic Ocean. His efforts to establish settlements on the island of Hispaniola, initiated the Spanish colonization of the New World [17]. Instead of reaching his intended place to find new trade during his first voyage in 1492, he landed in a New World.

**4.3 VIRGINIA DARE (FIRST CHILD BORN IN THE NEW WORLD)**

Virginia Dare was the first child born in the New World to English parents (Eleanor and Ananias Dare). She was born on August 18, 1587 and her date of death is unknown. Being the first Christian born in Virginia Colony, she was named after it [7]. The fact of her birth is known because of the Virginia Dare's grandfather, John White who was the governor of the settlement, returned to England in 1587 seeking fresh supplies.

**4.4 SALEM WITCH TRIALS**

Between February 1692 and May 1693, colonial Massachusetts’s prosecuted people accused of witchcraft; series of such hearings are called The Salem witch trials. Targeting mostly women, the trails executed twenty people [18]. It is also conducted in several other towns in the province of Massachusetts Bay like Ipswich and Andover, Salem Village (now Danvers).
4.5 Anne Bradstreet
Anne Bradstreet was born in March 20, 1612 and died September 16, 1672. She was one of the most prominent early English poets of North America and also the first female writer in the British North American colonies [19].

4.6 Seventeenth Century
Men and women who refused to compromise religious conviction and fled Europe in the face of European persecution settled most of the British North American Colonies that later formed the United States of America in seventeenth century. The New England, Pennsylvania, New Jersey and Maryland were some of the initially established colonies for plantations of religion [20].

4.7 Eighteenth Century
The 18th century- it is the second half of the colonial era in America. It was difficult for the colonial people to adjust at the very first and faced challenges but by the early 18th century things are getting better, they started living in houses as comfortable as those in Europe, rich people had comfortable chairs, wallpapers, beautiful carved furniture, also metal stove was invented by Benjamin Franklin in 1742 to make life more easy [21].
CHAPTER 5

LOOK AND FEEL OF THE ENTIRE APPLICATION

5.1 Home Page

Figure 5.1. Home page of the application.

Home page of the application includes Navigation bar at the top, which further consists of different menus to navigate to different pages. Below are the menus and submenus provided at the navigation bar.

- Maps
  - Submenus -> Norse Voyages
  - Columbus Voyages
  - Thirteen colonies
  - Custom Voyages
• Timeline
• Gallery

Flash image slideshow of vintage images is added at the middle section of the home page.

Some important colonization eras like Before Columbus, 1492-1600, Seventeenth century and Eighteenth century are added as section tiles at the bottom of the home page, which redirects to the respective pages on click.

5.2 Maps

Figure 5.2. Map 1: Columbus Voyages. Map showing Columbus voyages

Columbus voyages map is implemented to display the routes of the four voyages commanded by Columbus and sailed west to find a new trade route to the East but unexpectedly found the Americas.

Google polylines are used to display the routes on map, customized ship icon is added to make the starting and landing points more realistic, different path colors are used for different voyages to distinguish the voyages path and hover text is added on the path lines to explain the voyage name and it’s sailing path.

Tools like Clear map, previous voyage, next voyage, all voyages are provided to add more functionality to the interface and to make it more easy to use.
Figure 5.3. Map 2: Thirteen original colonies. Map showing thirteen colonies and three colonial regions

Original thirteen colonies map is implemented in the application using Google maps markers. Customized flag icons are used and hover text is added to make it informative.

Google map polygons are used to show three colonial regions by defining LatLong coordinates for colonial polygon’s path. Info windows are added to each colonial region to display detailed information about the regions on clicking the regional path.

Figure 5.4. Map 3: Norse voyages.
Norse voyages that are sailed from Norway to Iceland, Greenland and explored northern America are shown using Google polylines on the map.

Different path colors are used to distinguish the voyages and info text (hover text) is added to each voyages path to provide information about the voyage and its sailing route.

![Map showing Norse voyages](image)

**Figure 5.5. Map 4: Custom voyage interface.**

Interface is developed to create custom voyages; user with known longitude and latitude of start and end points can create a custom voyage using this tool.

This tool provides options to choose among different iconic paths like ship, aircraft, train, etc. and can be extended and embedded to build various visually interactive map applications.

Chapter 6 provides detailed description about custom voyages interface.

### 5.3 Timeline

Interactive timeline is built to show different colonization eras of American colonization history, an extensive research was done on American colonization history and only selected eras are presented on the timeline.

Slider is implemented to show different eras on timeline and different pages are added to show detailed information about specific era.
Figure 5.6. Timeline showing different American Colonial Eras.

Separate HTML pages of all the colonization era are added to provide detailed information about an era. Pages are linked to the timeline by providing back and forth navigation functionality.

5.4 Gallery

Figure 5.7. Application’s Gallery, includes books and videos.
Selective books and videos on American colonization history are added to the gallery. Gallery can be extended to add more books, documentaries and videos to provide reference to the user.
CHAPTER 6

MAP IMPLEMENTATION

This chapter describes the logical and customized implementation of the application’s map and the code used to embed the maps using Google maps APIs.

6.1. COLUMBUS VOYAGES MAP

![Figure 6.1. Columbus Voyages Map]

Columbus made four separate voyages across the Atlantic to the Americas, and his paths were similar. So one area of concern is how to represent those four voyages on a map.

One approach would be to make four separate “map layers”, one per voyage, or four separate layers. This has obviously disadvantages, as does single “static” maps showing all four voyages at once, which is at best “cluttered”. In any case, a more dynamic presentation is desirable in a world non filled with animation.

Google maps also allow for animation, different colored polylines are drawn on the map to show different voyages sailed to the west and their return route are also explained.
Code

```html
<!DOCTYPE html>
<html>
<head>
  <title>Columbus Voyages</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <style>html, body {
    height: 100%;
    width: 100%;
    margin: 0px 0px;
    padding: 0px
  }</style>

  // Embedding Google maps API

  <link rel="stylesheet" href="css/screen.css" type="text/css" media="screen">
  <link rel="stylesheet" href="css/responsive.css" type="text/css" media="screen">
  <link rel="stylesheet" href="css/colorbox.css" type="text/css" media="screen">
  <script src="jquery-2.1.1.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function () {
      $('#include').load("index.html");
    });
  </script>
</head>
<body>
  <style>
  #panel {
    position:absolute;
    top: 120px;
    left: 35%;
    margin-left: -50px;
    z-index: 1;
  }
</body>
</html>
```
background-color: cadetblue;
padding: 1px;
border: 1px solid #999;
-webkit-animation: mymove 5s infinite; /* Chrome, Safari, Opera */
animation: mymove 5s infinite;
}
</style>
</head>
<body>
<section class="header">
<div class="container">
<header>
<h1 class="navbar-brand navbar-brand_logo"><a href="index.html">AMERICAN COLONIZATION</a></h1>
<nav class="navbar navbar-default navbar-static-top my_navbar clearfix" role="navigation">
<ul class="nav sf-menu clearfix">
<li class="sub-menu"><a href="#">Maps</a><em></em></li>
<li><a href="timeline.html">Timeline</a><em></em></li>
<li><a href="gallery.html">Gallery</a><em></em></li>
</ul>
</nav>
</header>
</div>
</section>
<br/><br/>
// Buttons implementation
<div id="panel">
<button type="button" class="btn-native" onclick="clean()">Clear Map</button>
<button type="button" class="btn-native" onclick="prev()"><nbsp>&nbsp;Previous</button>
<button type="button" class="btn-native" onclick="next()">Next&nbsp;&nbsp;</button>
<button type="button" class="btn-native" onclick="showAll()">View All Voyages</button>
// Adding longitude and latitude details of four voyages in an array data structure along with information like voyage’s path color and info window.

var arrFunction = [
  // First Voyage data
  // "lat1": "38.7139", "long1": "-9.1394", "lat2": "22.403", "long2": ":79.75", "html": "test.html", "divId": "Voya", "Voyage": " First Voyage",
  "anicolor": "blue", "pathcolor": "steelblue"},
  // "lat1": "22.403", "long1": ":79.75", "lat2": "19.0000", "long2": ":70.6667", "html": "test.html",
  "divId": "Voya", "Voyage": " First Voyage",
  "anicolor": "blue", "pathcolor": "steelblue"},
  // "lat1": "19.0000", "long1": ":70.6667", "lat2": "37.23", "long2": ":6.90", "html": "test.html",
  "divId": "Voya", "Voyage": " First Voyage",
  "anicolor": "blue", "pathcolor": "steelblue"},
]
// Add other voyages

var myLatlng = new google.maps.LatLng(44.5403, -90.5463);
var mapOptions = {zoom: 3, center: myLatlng, streetViewControl: false,
   scrollwheel: false, navigationControl: false, mapTypeControl: false, scaleControl: false};

// Initializing markers to display path of voyages and landing marks.

var i = 0;
var markers = [];
var markers2 = [];
var flightPath = [];
function prev() {
   if (i > 0) removeLine(arrFunction[i]);
}
function clean() {
   removeLine1(arrFunction[i]);
}

var map = new google.maps.Map(document.getElementById('map-canvasC'), mapOptions);
map.setOptions({styles: styles});
var i = 0;
function next() {
   if (i < arrFunction.length) initialize(arrFunction[i], map);
}
function showAll() {
   removeLine1();
   for (i in arrFunction) {
      initialize(arrFunction[i], map);
   }
}

// Initializing maps

function initialize(map1, map) {
   var lat1 = map1.lat1;
   var lat2 = map1.lat2;
   var zoom = map1.zoom;

}
var long1 = map1.long1;
var long2 = map1.long2;
var pathcolor = map1.pathcolor;
var html = map1.html;
var divld = map1.divld;
var Voyage = map1.Voyage;
var date = map1.date;
var attack = map1.attack;
var namem1 = map1.namem1;
var namem2 = map1.namem2;
var anicolor = map1.anicolor;
$.removeClass("open");
$.addClass("closed");
myLatlng = new google.maps.LatLng(lat1, long1);
lineSymbol = {path: google.maps.SymbolPath.CIRCLE,
    icon: mimage, scale: 6, strokeColor: anicolor};
msymbol = {path: google.maps.Symbol, scale: 2};
p1 = new google.maps.LatLng(lat1, long1);
p2 = new google.maps.LatLng(lat2, long2);
flightPath[i] = new google.maps.Polyline({path: [p1, p2], geodesic: true,
    strokeColor: pathcolor, strokeOpacity: 1.0, strokeWeight: 2,
    icons:
        { icon: lineSymbol, offset: '10%'}
    });
mimage = 'Sailing_ship2.png';
marker = new google.maps.Marker({position: p1, map: map, title: namem1, icon: mimage});
marker2 = new google.maps.Marker({position: p2, map: map, title: namem2, icon: mimage});
markers.push(marker);
markers2.push(marker2);
infotext1 = null;
infowindow = null;
$.get(html, function (htmlr) { 
    $('#div1').append($(htmlr));
    divMain = "#" + divld;
    var htmlId = $('#div1').find(divMain).html();
    infowindow = new InfoBox({content: '<div class = "infoWindowOverflow">' + htmlId + '</div>'},
...
disableAutoPan: false, maxWidth: 0,
pixelOffset: new google.maps.Size(-115, 0), zIndex: null, boxClass: "myInfobox",
closeBoxMargin: "2px",
closeBoxURL: "close.png", infoBoxClearance: new google.maps.Size(1, 1), visible: true, pane: "floatPane", enableEventPropagation: false});

var textstring = '<div id="content" class = "infoWindowOverflow2" style="color: White; padding:3px; font-weight: bolder; font-size: 12px" id="firstHeading" class="firstHeading"><div></div><h4 style="margin-top: 5px">Voyage</h4>' + '
</div></div><h4 style="margin-top: 5px">Voyage</h4>' + '</div>;

infotext1 = new InfoBox({content: '<div class = "infoWindowOverflow2" style="background-color:orangered">' + textstring + '</div>', disableAutoPan: false, pixelOffset: new google.maps.Size(-140, 0), zIndex: null, boxClass: "myInfobox2", boxStyle: {
    opacity: 1, width: "250px" } });

google.maps.event.addListener(flightPath[i], 'click', function () {
    infowindow.open(map, marker2);
});
google.maps.event.addListener(flightPath[i], 'mouseover', function () {
    infotext1.open(map, marker2);
    var tempObj = this;
    tempObj.setOptions({strokeColor: "black"});
    this.strokeWeight = 8;
});
google.maps.event.addListener(flightPath[i], 'mouseout', function () {
    infotext1.close();
    var tempObj = this;
    tempObj.setOptions({strokeColor: pathcolor});
    this.strokeWeight = 4;
});
animateCircle(flightPath[i], map);
addLine();

// to set all markers
function setAllMap(map) {
    for (var i = 0; i < markers.length; i++) {
        markers[i].setMap(map);
    }
}
markers2[i].setMap(map);
}
}
// to add voyage path
function addLine() {
    flightPath[i].setMap(map);
    if (i < arrFunction.length) {
        i++;
    }
}

function addLine1() {
    while (i > 0) {
        i--;
        flightPath[i].setMap(null);
    }
}

function animateCircle(flightPath, map, infotext1) {
    flightPath.setMap(map);
    var count = 0;
    var timesRun = 0;
    var interval = setInterval(function () {
        timesRun += 1;
        if (timesRun === 199) {
            clearInterval(interval);
        }
        count = (count + 1) % 200;
        var icons = flightPath.get('icons');
        icons[0].offset = (count / 2) + '%';
        flightPath.set('icons', icons);
    }, 1);
}
6.2. THIRTEEN COLONIES AND THREE COLONIAL REGIONS (NEW ENGLAND, MIDDLE COLONIES, SOUTHERN COLONIES) MAPS

Figure 6.2. Thirteen colonies and three colonial regions.

The original thirteen colonies that were setup in the United States are divided into three colonial regions – New England, Middle Colonies and Southern Colonies.

The approach to show these on maps is by using Google map’s polygons to represent the three regions. When the user clicks on the polygon area i.e. a colonial region in our case, an info window opens, showing information about the region.

A Google map polygon takes longitude and latitude as input coordinates to form a polygonal region.

Thirteen colonies are represented on maps using Google map’s markers. One can use customized marker icons to display the related information accordingly; we have used colonial flag icons to display colonial marks on the map.
Code

<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <title>Maps</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Your description">
  <meta name="keywords" content="Your keywords">
  <meta name="author" content="Your name">
  <!--CSS-->
  <link rel="stylesheet" href="css/bootstrap.css">
  <link rel="stylesheet" href="css/style.css">
  <!--JS-->
  <script src="js/jquery.js"></script>
  <script src="js/jquery-migrate-1.2.1.js"></script>
  <script src="js/superfish.js"></script>
  <script src="js/jquery.easing.1.3.js"></script>
  <script src="js/jquery.mobilemenu.js"></script>
  <script src="js/jquery.cookie.js"></script>
  <script src="js/jquery.equalheights.js"></script>
  <script src="js/jquery.ui.totop.js"></script>

  <style>
    html, body, #map-canvas {
      width: 1500px;
      height: 700px;
      position: absolute;
      left: 0px;
      top: 0px;
    }
  
    #panel {

position: absolute;
top: 110px;
left: 45%;
margin-left: -150px;
z-index: 5;
background-color: cadetblue;
padding: 5px;
border: 1px solid #999;

-webkit-animation: mymove 5s infinite; /* Chrome, Safari, Opera */
animation: mymove 5s infinite;
}

/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
  50% {background-color: #3276b1;}
}
/* Standard syntax */
@keyframes mymove {
  50% {background-color: #086B95;}
}
</style>

// Using Google map API

<link rel="stylesheet" href="index1.css">
<script>

// Array structure designed to store data of the thirteen colonies

var citymap = {};
citymap['New York'] = {
  center: new google.maps.LatLng(40.7127, -74.0059),
text: '#000000',
text2: '#000000',
info: '<b>New York</b>'
}
citymap['New Hampshire'] = {
  center: new google.maps.LatLng(44.0000, -71.5000),
  text: 'black',
  text2: 'black',
  info: '<b>New Hampshire</b>'
};

citymap['Pennsylvania'] = {
  center: new google.maps.LatLng(41.0000, -77.5000),
  text: 'black',
  text2: 'black',
  info: '<b>Pennsylvania</b>'
} // Add other colonies data like above.

// Another array structure designed to store data of three colonial regions. 
var citymap2 = {};

citymap2['Middle Colonies'] = {
  triangleCoords: [
    new google.maps.LatLng(40.7127, -74.0059), //New York
    new google.maps.LatLng(41.31, -73.08), //New haven
    new google.maps.LatLng(44.29, -72.87), // New York
    new google.maps.LatLng(42.9047, -78.8494), //Buffalo
    new google.maps.LatLng(39.52, -81.03), // Delaware
    new google.maps.LatLng(40.0000, -74.5000), // New Jersey
  ],
  info: '<b>Middle Colonies</b><br/><a href="http://en.wikipedia.org/wiki/Middle_Colonies" target="_blank"><img src="MiddleColonies.gif" height="200" width="300"></a>',
  center: new google.maps.LatLng(39.0000, -75.5000)
} // Add other colonial region data like the above one.

// Map initialization, centered to specific long and lat.
var map;
var mapOptions;
function initialize() {
mapOptions = {
    zoom: 5,
    center: new google.maps.LatLng(42.5403, -90.5463)
};
map = new google.maps.Map(document.getElementById('map-canvas'),
    mapOptions);

showAll();
}
var markers = [];
var infowindows = [];
var taklaMakan = [];
var populationOptions = [];
var circle = [];
var yellow = [];
var image = null;
var pop = true;
var col = true;
var all = true;

function showColonialCities() {
    if (pop) {
        showColonialCities1(map);
        pop = false;
    }
    else {
        showColonialCities2(map);
        pop = true; }}

// Function to show colonies using Google maps marker.

function showColonialCities1() {
    for (var city in citymap) {
        image = new google.maps.MarkerImage(
            "PopK.png",
            null, /* size is determined at runtime */
            null, /* origin is 0,0 */
null, /* anchor is bottom center of the scaled image */
new google.maps.Size(40, 50)
);

markers[city] = new google.maps.Marker({
  position: citymap[city].center,
  map: map,
  icon: image,
  animation: google.maps.Animation.DROP
});
infowindows[city] = new google.maps.InfoWindow({
  maxWidth: 600,
  content: '<div>' + citymap[city].info + '<p>Population:</p>' + citymap[city].population + '</div>'
});
googel.maps.event.addListener(markers[city], 'mouseover', function (innerKey) {
  return function () {
    infowindows[innerKey].open(map, markers[innerKey]);
  }
})(city));
googel.maps.event.addListener(markers[city], 'mouseout', function (innerKey) {
  return function () {
    infowindows[innerKey].close();
  }
})(city));
populationOptions[city] = {
  strokeColor: citymap[city].text,
  title: citymap[city].info,
  strokeOpacity: 0.6,
  strokeWeight: 2,
  fillColor: citymap[city].text2,
  fillOpacity: 0.15,
  map: map,
  center: markers[city].position,
  radius: 30000
};
circle[city] = new google.maps.Circle(populationOptions[city]); })}
function showColonialCities2() {
    for (var city in citymap) {
        markers[city].setMap(null);
        circle[city].setMap(null); }

// Function to show colonial regions

function showColonies() {
    if (col) {
        showColonies1(map);
        col = false;
    } else {
        showColonies2(map);
        col = true; }

function showColonies1() {
    for (var city in citymap2) {
        markers[city] = new google.maps.Marker({
            position: citymap2[city].center,
            map: map,
            visible: false
        });

        taklaMakan[city] = new google.maps.Polygon({
            paths: citymap2[city].triangleCoords,
            strokeColor: 'darkgreen',
            strokeOpacity: 0.8,
            geodesic: true,
            strokeWeight: 1.0,
            fillColor: 'yellow',
            fillOpacity: 0.35,
            map: map
        });

        infowindows[city] = new google.maps.InfoWindow({
            content: citymap2[city].content
        });
    }}
maxwidth: 400,
content: '<div>' + citymap2[city].info + '</div>
});

google.maps.event.addListener(taklaMakan[city], 'click', function (innerKey) {
    return function () {
        infowindows[innerKey].open(map, markers[innerKey]);
    }
}(city));

google.maps.event.addListener(taklaMakan[city], 'mouseover', function (innerKey) {
    return function () {
        taklaMakan[innerKey].setOptions({strokeWeight: 4});
        taklaMakan[innerKey].setOptions({strokeColor: "darkgreen");
        taklaMakan[innerKey].setOptions({fillOpacity: 0.8});
    }
}(city));

google.maps.event.addListener(taklaMakan[city], 'mouseout', function (innerKey) {
    return function () {
        taklaMakan[innerKey].setOptions({strokeWeight: 0.8});
        taklaMakan[innerKey].setOptions({strokeColor: "darkgreen");
        taklaMakan[innerKey].setOptions({fillOpacity: 0.35});
    }
}(city));

// to show colonial regions
function showColonies2() {
    for (var city in citymap2) {
        taklaMakan[city].setMap(null); }}

// Functions used to trigger functionalities of buttons provided at tool tab.

function showAll() {
    if (col == false) {
        showColonies2();
    }
    if (pop == false) {
        showColonialCities2();
    }
if (all) {
    showAll1();
    all = false;
    pop = false;
    col = false;
}
else {
    showAll2();
    all = true;
    pop = true;
    col = true; }}

function showAll1() {
    showColonies1();
    showColonialCities1();
}
function showAll2() {
    showColonies2();
    showColonialCities2();
}

google.maps.event.addDomListener(window, 'load', initialize);
</script>
</head>
<body>
<div id="panel">
<button onclick="showColonies(map)">Show Colonies</button>
<button onclick="showColonialCities(map)">Show Regions</button>
<button onclick="showAll(map)" class="my_fsm">Show All</button>
</div>
<section class="header">
<div class="container">
<header>
<h1 class="navbar-brand navbar-brand_logo"><a href="#">AMERICAN COLONIZATION</a></h1>
<nav class="navbar navbar-default navbar-static-top my_navbar clearfix" role="navigation">
6.3. ADDING CUSTOM PATHS TO THE MAP

Figure 6.3. Adding custom voyages to the map.

Providing a flexible tool to let user add the required information in most visually interactive way has always been a great addition to any application.

To enhance the Geographical Information System, an interface is developed where a user can display information on map by providing initial and destination longitudinal
latitudinal points and can customize the informational path (like icons, path color, etc.) according to the needs.

Tool usage:

- Enter latitude and longitude of start and end point in text boxes: Start point Lat, Start point Long, End point Lat, End point Long.
- Select marker icons from dropdown list.
- Use ‘Clear’ button to clear the text boxes.
- Use ‘Next’ button to display the custom path.
- Use ‘Clear Map’ to remove the path from the map.
- Use the ‘+’, ‘-’ for zoom control on the map.

The following code with added detailed comments shows the details of how the custom voyage tool implementation was achieved.

```html
<!DOCTYPE html>
<html>
<head>
  <title>Columbus Voyages</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <style>html, body {
    height: 100%;
    width: 100%;
    margin: 0px 0px;
    padding: 0px
  }
</style>

// Embedding google map api
  <link rel="stylesheet" href="css/screen.css" type="text/css" media="screen">
  <link rel="stylesheet" href="css/responsive.css" type="text/css" media="screen">
  <link rel="stylesheet" href="css/colorbox.css" type="text/css" media="screen">
  <script src="jquery-2.1.1.min.js"></script>
```
<script type="text/javascript">
    $(document).ready(function () {
        $('#include').load("index.html");
    });
</script>

<!--CSS-->
<link rel="stylesheet" href="css/bootstrap.css">
<link rel="stylesheet" href="css/style.css">

<style>
    #panel {
        position:absolute;
        top: 130px;
        left: 45%;
        margin-left: -50px;
        z-index: 1;
        background-color: cadetblue;
        padding: 1px;
        border: 1px solid #999;
        -webkit-animation: mymove 5s infinite; /* Chrome, Safari, Opera */
        animation: mymove 5s infinite;
    }
</style>

<style>
    #customTool {
        position:absolute;
        top: 200px;
        left: 10%;
        margin-left: -50px;
        z-index: 1;
        background-color: darkcyan;
        padding: 1px;
        border: 1px solid #999;
        -webkit-animation: mymove 5s infinite; /* Chrome, Safari, Opera */
        animation: mymove 5s infinite;
    }
</style>
<form>
    <label for="StartLat">Start point Lat: </label><input type="text" name="StartLat" id="StartLat"/>
    <label for="StartLong">Start point Long: </label><input type="text" name="StartLong" id="StartLong"/>
    <label for="EndLat">End point Lat: </label><input type="text" name="EndLat" id="EndLat"/>
    <label for="EndLong">End point Long: </label><input type="text" name="EndLong" id="EndLong"/>
</form>

// Dropbox
<form>
    Select your type:
    <select id="mySelect">
        <option value="ship">Ship</option>
        <option value="aircraft">Aircraft</option>
        <option value="train">Train</option>
    </select>
</form>
</select>
</form>
<br class="clear" />
<input type="button" onclick="clearFields() , clean(), clearMarkers()" value="Clear">
</form>
</div>

// To clear text forms
<script type="text/javascript">
function clearFields() {
    document.getElementById("StartLat").value="";
    document.getElementById("StartLong").value="";
    document.getElementById("EndLat").value="";
    document.getElementById("EndLong").value="";
}
</script>

<section class="header">
<div class="container">
<header>
<h1 class="navbar-brand navbar-brand_logo"><a href="index.html">AMERICAN COLONIZATION</a></h1>
<nav class="navbar navbar-default navbar-static-top my_navbar clearfix" role="navigation">
<ul class="nav sf-menu clearfix">
<li class="sub-menu"><a href="#">Maps</a><em></em></li>
<li><a href="timeline.html">Timeline</a><em></em></li>
<li><a href="gallery.html">Gallery</a><em></em></li>
</ul>
</nav>
</header>
</div>
</section>
<br/><br/>
<div id="panel">
<button type="button" class="btn-native" onclick="clean()">Clear Map</button>
<button type="button" class="btn-native" onclick="prev()"> << Previous</button>
</div>
<button type="button" class="btn-native" onclick="next()">Next »</button>

<!--<button type="button" class="btn-native" onclick="showAll()">View All Voyages</button>-->
</div>

<div id="map-canvasC"
    style="width: 1450px; height: 700px; position: absolute; left: 0px; top: 0px;"
    >
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script type="text/javascript" src="css/colorbox.js"></script>
<script type="text/javascript" src="js/timeliner.js"></script>
<script> $(document).ready(function () {
    $.timeliner({});
    $('.CBmodal').colorbox({inline: true, initialWidth: 100, maxWidth: 682, initialHeight: 100, transition: "elastic", speed: 750});
});</script>

<script>
    var arrFunction = [
        {"lat1": document.getElementById('StartLat').value , "long1": document.getElementById('StartLong').value, "lat2": document.getElementById('EndLat').value, "long2": document.getElementById('EndLong').value , "html": "testDisplay.html", "divId": "Voyage", "Voyage": " Custom Voyage", "anicolor": "Red", "pathcolor": "Red"},
    ];
    var myLatlng = new google.maps.LatLng(64.5403, -45.5463);
    var mapOptions = {zoom: 3, center: myLatlng, streetViewControl: false,
        scrollwheel: false, navigationControl: false, mapTypeControl: false, scaleControl: false};
    var i = 0;
    var markers = [];
    var markers2 = [];
var flightPath = [];

function prev() {
    if (i > 0) removeLine(arrFunction[i]);
}

// to remove markers and path
function clean() {
    removeLine(arrFunction[i]);
    clearMarkers();
}

var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
map.setOptions({styles: styles});
var i = 0;

// to display the voyage path
function next() {
    if (i < arrFunction.length) initialize(arrFunction[i], map);
}

function showAll() {
    removeLine1();
    for (i in arrFunction) {
        initialize(arrFunction[i], map);
    }
}

// Initializing the map and markers
function initialize(map1, map) {
    var lat1 = document.getElementById('StartLat').value;
    var lat2 = document.getElementById('EndLat').value;
    var zoom = map1.zoom;
    var long1 = document.getElementById('StartLong').value;
    var long2 = document.getElementById('EndLong').value;
    var pathcolor = map1.pathcolor;
    var html = map1.html;
    var divld = map1.divld;
    var Voyage = map1.Voyage;
    var date = map1.date;
    var attack = map1.attack;
var namem1 = map1.namem1;
var namem2 = map1.namem2;
var anicolor = map1.anicolor;

// To select icons according to dropdown option select
if(document.getElementById("mySelect").value == "ship")
{
    var mimage = "norse4.png";
}
else if(document.getElementById("mySelect").value == "aircraft")
{
    var mimage = "Plane.png";
}
else{
    var mimage = "Train2.png";
}

$(".timelineMinor a").removeClass("open");
$(".timelineMinor a").addClass("closed");
var myLatlng = new google.maps.LatLng(lat1, long1);
var lineSymbol = {path: google.maps.SymbolPath.CIRCLE,
    icon: mimage, scale: 6, strokeColor: anicolor};
var msymbol = {path: google.maps.Symbol, scale: 2}
var p1 = new google.maps.LatLng(lat1, long1);
var p2 = new google.maps.LatLng(lat2, long2);
flightPath[i] = new google.maps.Polyline({ path: [p1, p2], geodesic: true,
    strokeColor: pathcolor, strokeOpacity: 1.0, strokeWeight: 2,
    icons: [
        { icon: lineSymbol, offset: '10%'}
    ] });

var marker = new google.maps.Marker({position: p1, map: map, title: namem1, icon: mimage})
var marker2 = new google.maps.Marker({position: p2, map: map, title: namem2, icon: mimage})
markers.push(marker);
markers2.push(marker2);
var infotext1 = null;
var infowindow = null;
$.get(html, function (htmlr) {
    $('#div1').append($(htmlr));
    divMain = "#" + divId;
    var htmlId = $('#div1').find(divMain).html();
infowindow = new InfoBox({content: '<div class = "infoWindowOverflow">' + htmlId + '</div>',
        disableAutoPan: false, maxWidth: 0,
        pixelOffset: new google.maps.Size(-115, 0), zIndex: null, boxClass: "myInfobox",
        closeBoxMargin: "2px",
        closeBoxURL: "close.png", infoBoxClearance: new google.maps.Size(1, 1), visible: true,
        pane: "floatPane", enableEventPropagation: false});

    var textstring = '<div id="content" class = "infoWindowOverflow2" style="color: White;
        padding:3px; font-weight: bolder; font-size: 12px" id="firstHeading" class="firstHeading"><div
></div><h4 style="margin-top: 5px">Voyage</div>' + Voyage + '</div>;

    infotext1 = new InfoBox({content: '<div class = "infoWindowOverflow2" style="background-color:orangered">' + textstring + '</div>', disableAutoPan: false, pixelOffset: new google.maps.Size(-140, 0), zIndex: null, boxClass: "myInfobox2", boxStyle: {
        background: "url(http://google-maps-utility-library-v3.googlecode.com/svn/trunk/infobox/examples/tipbox.gif) no-repeat",
        opacity: 1, width: "250px" }});

    google.maps.event.addListener(flightPath[i], 'click', function () {
        infowindow.open(map, marker2);
    });

    google.maps.event.addListener(flightPath[i], 'mouseover', function () {
        infotext1.open(map, marker2);
        var tempObj = this;
        tempObj.setOptions({strokeColor: "black"});
        this.strokeWeight = 8;
    });

    google.maps.event.addListener(flightPath[i], 'mouseout', function () {
        infotext1.close();
        var tempObj = this;
        tempObj.setOptions({strokeColor: pathcolor});
        this.strokeWeight = 4;
    });
});

animateCircle(flightPath[i], map);
addLine();
}

// To set all the map's markers
function setAllMap(map) {
    for (var i = 0; i < markers.length; i++) {
        markers[i].setMap(map);
        markers2[i].setMap(map);
    }
}

// To add voyage path
function addLine() {
    flightPath[i].setMap(map);
    if (i < arrFunction.length) {
        i++;
    }
}

// to remove voyage path
function removeLine() {
    i--;
    flightPath[i].setMap(null);
}

function removeLine1() {
    while (i > 0) {
        i--;
        flightPath[i].setMap(null);
    }
}

// to remove markers
function clearMarkers() {
    setAllMap(null);
}

function animateCircle(flightPath, map, infotext1) {
    flightPath.setMap(map);
    var count = 0;
    var timesRun = 0;
    var interval = setInterval(function () {

timesRun += 1;
if (timesRun === 199) {
    clearInterval(interval);
}
count = (count + 1) % 200;
var icons = flightPath.get('icons');
icons[0].offset = (count / 2) + '%';
flightPath.set('icons', icons);
}, 1);
}
</script>
</div>
</body>
</html>
CHAPTER 7

SUMMARY, OBSTACLES, FUTURE WORK, AND TESTING

During this project, I came to know so many interesting facts about Google map API and came to know about the colonization of America, how Europeans came in and colonized the parts of America, first initially setting up thirteen colonies, how Columbus sailed his four voyages west to find a new trade route to the East but unexpectedly found the Americas.

It was an excellent learning practice to build a map-based application using JavaScript, HTML, CSS3 and embedded Google maps using Google maps API. The biggest challenge was to show voyages on map in most visually compelling way for which Google map polylines and markers were chosen. Thirteen original colonies and colonial regions are shown using Google map polygons. Major hurdle was to make the application more interactive for which an interface is developed which let user to create their own paths on map, this interface can be extended to various application in future to show dynamic paths like air paths, train circuits, etc. Also there were minor hurdles in collecting old vintage-look images to make the application more attractive.

The application provides vast information about the colonization of America and has interactive tools to learn more about it.

Future Work: Providing different language options to change entire application’s language to other language like Spanish, etc. would be a great addition to the application, it will make the application more users friendly. Making the application more mobile/tab friendly in future can further advance the application.

The Application was tested by several people (friends and co-workers), also by people who are not from a computer science background. Below are the changes and suggestions suggested by them:
- More icon options at custom interface tool.
- Flash displaying images added at the landing page.
- Separate HTML pages added for American colonial Eras.
- Different colors and icons added for voyages.
- Navigation bar added to each page.
REFERENCES


