Design and develop cutting-edge mobile web applications using jQuery Mobile to work across a number of platforms

What you will learn from this book
- Create a theme for your jQuery Mobile application by using the ThemeRoller tool
- Test your application on a variety of resolutions through Screenfly
- Master pages, panels, popups, toolbars, and navbars in jQuery Mobile
- Understand how mobile web design best practices can enhance your mobile web application
- Allow your users to navigate pages with a swipe gesture and add transitions and effects to the page changes
- Discover how to create jQuery Mobile widgets that will extend the framework to suit your needs
- See how Apache Cordova can turn your web application into a native Android or iOS app
- Use jQuery Mobile with other popular JavaScript frameworks and technologies such as Backbone.js and Node.js

Who this book is written for
You've started down the path of jQuery Mobile, now begin mastering some of jQuery Mobile's higher level topics. Go beyond jQuery Mobile's documentation and master one of the hottest mobile technologies out there. Previous JavaScript and PHP experience can help you get the most out of this book.
In this package, you will find:

- The author’s biography
- A preview chapter from the book, Chapter 4 ‘Call to Action – Our Main Project’
- A synopsis of the book’s content
- More information on Mastering jQuery Mobile
About the Authors

**Chip Lambert** has been creating websites ever since his high school days, when he started with fantasy sport websites. In addition to these 20 years of HTML experience, he has 15 years of experience with PHP and MySQL development. He previously wrote *Instant REST Implementation How-to* by Packt Publishing. He is currently a technical analyst for Jenzabar, Inc. You can follow him on Twitter at @chiplambert or visit his personal blog cum website at http://www.programmerchip.com.

**Shreerang Patwardhan** completed his bachelor's degree in computer engineering, and since then, he has been working on various technologies for the last 4.5 years. He started off by working for a small start-up in Pune, India, on an innovative solar-powered Linux-based handheld device. He has also worked extensively on Google Maps API v3 and worked in the GIS domain for more than a year. He is also one of the technical reviewers for *Google Maps JavaScript API Cookbook*, published by Packt Publishing.

Shreerang is currently employed at a MNC in the San Francisco Bay Area, USA, as a technical consultant and is working on the frontend development of various web applications using different cutting-edge frontend technologies. He is also a certified Scrum Master and absolutely loves and encourages the Agile methodology.

When not working on a project, he blogs about either the Google Maps API v3 or the jQuery Mobile framework on his blog, *Spatial Unlimited*. When not working or blogging, Shreerang loves spending time with his family and friends and absolutely enjoys sweating it out on the badminton court. He has been playing badminton for the last 20 years and also takes a keen interest in UFOlogy.

You can reach him on his blog, contact him on LinkedIn, or follow him on Twitter (@shreerangp).
jQuery Mobile really ups the ante for us mobile developers. It gives us a powerful framework to develop web applications tailored for mobile devices and have them act like native applications. We hope to take your jQuery Mobile skills to the next level with this book by having you develop projects that you may encounter in your everyday life, instead of a bunch of one-shot conceptual projects. We have one main project that we will build over a series of chapters and then some standalone projects that will explore some more advanced features.

What this book covers

Chapter 1, Getting Started, in this chapter you will begin setting up your development environment to be able to complete the project accompanying the book. You will learn the various methods to obtain and use the framework in a small sample application.

Chapter 2, Tools and Testing, in this chapter you will look at some additional tools that you can use in jQuery Mobile development. Apart from the different tools, this chapter will also look at some means of testing that will assist you in making sure your projects work well across all platforms.

Chapter 3, Mobile Design, in this chapter you will get a brief overview of developing for mobile devices. The topics will cover responsive web design techniques, including media queries, features, and device detection, and will take a very brief look at RESS.

Chapter 4, Call to Action – Our Main Project, in this chapter you will begin the development of our overall project, a mobile application for a fictional Anytown Civic Center. The application will display a list of upcoming events, allow users to enquire for the events, and serve as a mobile website for the civic center itself.
Preface

Chapter 5, Navigation, with this chapter, you will start putting together a touch-based navigation system as we link the pages together. You will also get to see the use of transitions.

Chapter 6, Controls and Widgets, in this chapter, you will explore the various widgets and controls that we will use to build the Anytown Civic Center application. You will learn how to configure and initialize widgets, and control the app with various input events.

Chapter 7, Working with Data, in this chapter, you will look at how we can retrieve data using PHP and use it within our mobile application. We will look at forms, retrieving data from a database, validating input before inserting it back into the database, and displaying information to the application user.

Chapter 8, Finishing Touches, in this chapter, you will finish up the Anytown Civic Center mobile application by creating custom icons to use in the application to replace the default ones. We will also begin exploring how this application can be turned into a native mobile application with a basic introduction to Cordova.

Chapter 9, The Next Level, in this chapter, you will look at using jQuery Mobile with the Node.js platform, integrating jQuery Mobile with the Backbone.js and RequireJS libraries, and build a basic WordPress mobile theme that will use this framework.

Chapter 10, Mobile Best Practices and Efficiency, in this chapter, you will look at some best mobile practices and some optimization tips and tricks that will benefit you as you go deeper into jQuery Mobile development.
We are firm believers in the fact that the best way to learn something, especially a programming topic, is through hands-on, practical examples. So, this book will be no different. We are going to build a mobile web application for Anytown Civic Center. It will display events going on at the civic center, allow users to register for these events, and serve as a general mobile website for the civic center.

Overview

It is time to get down to business now. This project will display a list of upcoming events to the visitors, and allow them to register for the events that interest them. It will also have general information about the civic center on it, such as hours, driving directions, and more.

This project will be made using several features, or as they are known in jQuery Mobile: roles that make up the framework. These roles include pages, panels, popups, toolbars, and navbars.

As we go through this project, you may notice that jQuery Mobile uses the `<div>` element quite a bit instead of using, for example, HTML5's `<header>` and `<footer>` elements. This is to ensure backwards-compatibility with browsers that may not support HTML5.
Pages

Pages are the main unit within jQuery Mobile. The jQuery Mobile framework can support single pages or multiple internally-linked pages within one page (this may sound confusing right now, but you'll see shortly).

Pages themselves are made up of three components: header, content, and footer. Remember, as we mentioned previously, jQuery Mobile does not use the common HTML5 structure elements, so you will instead use these header, content, and footer implementations so that you can ensure the most cross-browser compatibility as you can.

Let's see an example of a page:

```html
<div data-role="page">
  <div data-role="header">
    <!-- Header stuff -->
  </div><!-- /header -->

  <div role="main" class="ui-content">
    <!-- Content -->
  </div><!-- /content -->

  <div data-role="footer">
    <!-- Footer stuff -->
  </div><!-- /footer -->

</div><!-- /page -->
```

As you'll see throughout this chapter, whenever you start a new role, you use `<div data-role=""`.

One very important note: in order to make the most of jQuery Mobile, you must start the HTML page itself with the HTML5 doctype. Without it, you won't be able to take advantage of jQuery Mobile's power. Browsers that do not support HTML5 will simply ignore this doctype and continue working.

Let's go ahead and start working on our project. Create a new web project in Apatana Studio. I named mine anytowncc; you can use this name or choose another.

Now that the project is created, add a new a file to the project and name it `index.html` (if one was not already created for you by Aptana Studio). Add the following code to this file:

```html
<!DOCTYPE html>
<html>
```
We have seen all this code previously, so we will not break it down as we normally do. We declared our HTML5 doctype, we are loading in the jQuery and jQuery Mobile files from the jQuery CDN, and then we are creating our first jQuery Mobile page with a bit of placeholder text in it for right now.

One thing that we haven't used yet is `data-theme="a"`. You could add a `data-theme` attribute on the page level to style the entire page, based on the colors of a particular swatch you want. You could use one of the six default swatches, or you could even make use of the custom swatch that we created using the jQuery Mobile ThemeRoller tool.
Now, you can open this page that we just created up in your normal web browser, or even better, you can see what it would be like on a mobile device via one you have laying around or use Screenfly. No matter which method you use, you should see something like this:

Now, we have the first page ready. We'll be adding more later on, but for now we are going to move on to panels.
Panels

jQuery Mobile has fantastic support for all kinds of different panels. You can use them to create menus, inspector panes, and so much more. You can have them attached to the left or right-hand side of the page and you have three options for them:

- **reveal**: This panel sits beneath the page and reveals as the page slides away
- **overlay**: This panel will appear on top of the page
- **push**: This panel and the page will be animated at the same time

When creating panels, they must be created within a page, as a sibling to the header, content, and footer components. The panel can be either before or after these elements, but not in between them. The source order does not really matter as long as the preceding constraints are met and as long as your application is not meant for old devices that support just plain HTML, or your end users are not going to rely on screen readers. If your application needs to support the C-level devices and/or users rely on screen readers, then you would want to consider where you place the panel markup, as this would be read out to the users. There is talk that this restriction could be removed in a future version of the framework, but at the time of writing this book, this constraint must be followed.

Let’s add a panel to our page. We believe that the best place for the panel is after the header and before the content. This way, we can even account for users relying on screen readers and support users using the C-level devices. We will use the panel to hold our contact information. Open up the file we were just working on, if you have closed it, and add the following code after our footer div element:

```html
<div data-role="panel" id="contactpanel" data-display="push" data-dismissible="true" data-theme="a">
  <div>
    <p>Contact Us!</p>
    <a href="tel:555-555-5555">(555) 555-5555</a>
    <p>1234 First Avenue</p>
    <p>Anytown, Anystate 12345</p>
    <a href="mailto:contact@anytownciviccenter.com">
      contact@anytownciviccenter.com
    </a>
  </div>
</div><!-- /panel -->
```
Your file will now look like this:

```html
<!DOCTYPE html>
<html>
<head>
<title>Anytown Civic Center</title>
<meta name="viewport" content="width=device-width, 
   minimum-scale=1.0,maximum-scale=1.0, 
   user-scalable=no">
<link rel="stylesheet" 
   href="http://code.jquery.com/mobile/1.4.5/ 
   jquery.mobile-1.4.5.min.css" />
<script src="http://code.jquery.com/ 
   jquery-1.11.0.min.js"></script>
<script 
   src="http://code.jquery.com/mobile/1.4.5/ 
   jquery.mobile-1.4.5.min.js"></script>
</head>
<body>
<div data-role="page">
   <div data-role="header">
      <h1>Welcome to Anytown Civic Center!</h1>
   </div><!-- /header -->
   <div data-role="panel" id="contactpanel" 
      data-display="push" data-dismissible="true" 
      data-theme="a"> 
      <div>
         <p>Contact Us!</p>
         <a href="tel:555-555-5555">(555)555-5555</a><p>
         <p>1234 First Avenue</p>
         <p>Anytown, Anystate 12345</p>
         <a href="mailto:contact@anytownciviccenter.com" 
            contact@anytownciviccenter.com</a>
      </div>
   </div><!-- /panel -->
   <div class="ui-content" role="main">
      <p>Welcome to the website for Anytown Civic 
         Center. We have a lot of great upcoming events 
         for you to check out.</p>
   </div><!-- /content -->
</div>
</body>
</html>
```
Let's look at the code we added:

```html
<div data-role="panel" id="contactpanel" data-display="push"
    data-dismissible="true" data-theme="a">
</div>
```

We are declaring our panel and giving it the ID of `contactpanel`. We are telling the framework that we want this panel to be a push panel and be styled by the swatch of our theme, which happens to be the default theme that ships with jQuery Mobile.

Now, if you run this code, the page will look the same. "Why is that?" you ask. Well, we have to provide a means to make the panel visible. It remains offscreen until we execute the code to make it visible. You can do a couple of things to make it visible. You can use a link with the `href` set to the `id` element of `div` of the panel, or you can use a button to make it visible. The button looks better in our opinion, so this will be the method we will use. Change the footer section to look like the following code:

```html
<div data-role="footer">
    <h2>
        <a href="#contactpanel" class="ui-btn" data-rel="panel">Contact us</a>
    </h2>
</div>
```

What we are doing here is creating a jQuery Mobile button with the link pointing to our panel `div` element. Now, execute the code and push the button, and you should see the following screenshot:

Contact Us!

(555)555-5555

1234 First Avenue

Anytown, Anystate 12345

contact@anytownciviccenter.com
Looking good so far. To close the panel, you can add another button to the panel
<code>div</code> element that closes it, or you can simply click on the page itself to go back to
the main website. We'll be adding more to this panel as we progress through the
project; for now, let's move on to popups.

**Popups**

Popups have been around for quite some time in one aspect or another. jQuery
Mobile provides many implementations of popups that we can make use of.
These different types of popups include:

- Basic pop up
- Image lightbox
- Tooltip
- Modal dialog

In addition to these types of popups, the basic popup can be modified to hold menus,
maps, and more. jQuery Mobile does an excellent job of providing a very powerful
and robust popup system.

Over the course of this book, we will make use of a couple of the different types of
popups, but for now, let's make a modal dialog letting the user know that we don't
have any current events scheduled for them to register for.

Declaring one event is just as straightforward as the other jQuery Mobile widgets
we've already looked at. To create a popup, add the data-role of popup to a
<code>div</code> element that holds the pop-up contents. A link needs to be created with the
<code>href</code> set to the id element of the popup, <code>div</code>, and add the data-rel attribute with the
value popup. A popup, <code>div</code>, has to be nested inside the same page as that of the link.

Open up <code>index.html</code> and add the following code after our page declaration and
before our header declaration:

```html
<div data-role="popup" id="eventregister"
data-overlay-theme="b" data-theme="b"
data-dismissible="false">
  <div data-role="header" data-theme="a">
    <h1>Current Events</h1>
  </div>
</div>
```
<div role="main" class="ui-content">
<h3 class="ui-title">Sorry, currently there are no scheduled events.</h3>
<a href="#" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b" data-rel="back">Close</a>
</div>

Now, we need to code in some way to make our popup appear. We'll use another button to do this; add the following code in our current content area after `<p>Welcome to the website for ...:</p>`

`<a href="#eventregister" data-rel="popup" data-position-to="window" data-transition="pop" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b">View Current Events</a>`

Now, if you launch the website, you will first see this:
Looking good so far. Now, click on the View Current Events button and you will see this:

Awesome! We just created a very simple, yet elegant modal dialog box with jQuery Mobile. Let's take a look at the code to see how it was done:

```html
<div data-role="popup" id="eventregister" data-overlay-theme="b" data-theme="b" data-dismissible="false">
    Here, we are creating our popup and giving it the ID of eventregister. We are assigning it the b swatch of our current theme. By setting the data-dismissible option to false, we are telling the framework that the only way to get rid of the popup is to use the button on the modal itself. If we had set that to true, simply touching anywhere on the screen or using the Esc key would dismiss it:

    <div data-role="header" data-theme="a">
        <h1>Current Events</h1>
    </div>

    Here, we are creating the header part of our popup. You may be saying right now, "Hey, we already have a header" and you would be right. That's the good thing about the jQuery framework; we can set and use different headers for our widgets. If we didn't have the header set here, Current Events would not be stylized in the popup and it wouldn't look right. Go ahead and remove the header declaration (the data-role="header" piece) for now, and this is what you will get:

```
```
Looks horrible, doesn't it? Put the header code back and let's continue looking at the code:

```
<div role="main" class="ui-content">
  <h3 class="ui-title">Sorry, currently there are no scheduled events.</h3>
  <a href="#" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b" data-rel="back">Close</a>
</div>
```

This code is the content of our popup. We are applying some of jQuery Mobile's built-in CSS styling to a few of the elements, and then lastly, we will create a button that will be used to close the popup. Remember that this is the only way we can close the popup, since we set the `data-dismissible` attribute to `false`.

That's it for the popup itself, so now let's look at the button we will use to open it:

```
<a href="#eventregister" data-rel="popup" data-position-to="window" data-transition="pop" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b">View Current Events</a>
```

Here, we are creating a jQuery Mobile button and passing the `href` element the ID of the `div` element it needs to open. You can have several popups on a page, so please be sure to keep them straight with your `div` IDs. For `data-transition`, we are telling it to pop onto the screen, which will have it appear extremely fast with little fanfare.
Data transitions

There are 10 different data transitions that you can use on popups. These transitions are as follows:

- No transition
- Pop
- Fade
- Flip
- Turn
- Flow
- Slide
- Slide fade
- Slide up
- Slide down

Each of them has a different effect when bringing the popup; play around with the one we just created and try out these different transitions, so you can see each of them in action and find the one that you may like.

Here is what our overall code should now look like so far; in this way, if something isn't working for you anymore, you can refer to the code to make sure everything has been done correctly:

```html
<!DOCTYPE html>
<html>
<head>
<title>Anytown Civic Center</title>
<meta name="viewport" content="width=device-width, minimum-scale=1.0,maximum-scale=1.0, user-scalable=no">
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css" />
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</head>
<body>
<div data-role="page">
    <div data-role="popup" id="eventregister" data-overlay-theme="b" data-theme="b" data-dismissible="false">

```
<div data-role="header" data-theme="a">
    <h1>Current Events</h1>
</div>

<div role="main" class="ui-content">
    <h3 class="ui-title">Sorry, currently there are no scheduled events.</h3>
    <a href="#" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b" data-rel="back">Close</a>
</div>

<div data-role="header">
    <h1>Welcome to Anytown Civic Center!</h1>
</div><!-- /header -->

<div data-role="panel" id="contactpanel" data-display="push" data-dismissible="true" data-theme="a">
    <div>
        <p>Contact Us!</p>
        <a href="tel:555-555-5555">(555)555-5555</a>
        <p>1234 First Avenue</p>
        <p>Anytown, Anystate 12345</p>
        <a href="mailto:contact@anytownciviccenter.com">contact@anytownciviccenter.com</a>
    </div>
</div><!-- /panel -->

<div class="ui-content" role="main">
    <p>Welcome to the website for Anytown Civic Center. We have a lot of great upcoming events for you to check out.</p>
    <a href="#eventregister" data-rel="popup" data-position-to="window" data-transition="pop" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b">View Current Events</a>
</div><!-- /content -->

<div data-role="footer">
    <h2><a href="#contactpanel" class="ui-btn ui-corner-all" data-rel="panel">Contact us</a></h2>
</div>
Now that we are done with popups and everything appears to be working, let's move on to our next section.

**Toolbars**

Toolbars in jQuery Mobile are used to enhance the headers and footers of the pages. We have already used headers and footers in our page, so now it is time to make them a bit fancier and add some functionality to our app.

Toolbars is probably one of the easiest jQuery Mobile widgets to implement, especially if you already have a header or footer. To show how easy this is, we are going to change our existing header and footer sections into toolbars.

Before we get started, let's jot down our plan:

1. Create a **Home** button in our header.
2. Move our **Events** button from the content to our header.
3. Stylize our **Contact** button in our footer to look like a toolbar button.

This doesn't seem like a whole lot of stuff to do, but it will make a beautiful difference in our mobile app. To begin, we'll create a **Home** button.

**Creating a Home button**

Keeping with the trend, we've been going with on this chapter, open up `index.html` and add the following code right under our initial header declaration:

```html
<a href="index.html" class="ui-btn-left ui-btn ui-btn-inline ui-mini ui-corner-all ui-btn-b ui-btn-icon-left ui-icon-home">Home</a>
```
Save the code and view it, you should see this:

```
<a href="index.html" class="ui-btn-left ui-btn ui-btn-inline
ui-mini ui-corner-all ui-btn-b ui-btn-icon-left
ui-icon-home">Home</a>
```

We are creating a link to our main page, index.html (for now) by adding a button that has an icon in it. We have already seen how we style the link tag like a button using the jQuery Mobile specific button classes in the example so far. A few new classes here include:

- **ui-btn-left**: This class informs the framework that the button is to be aligned to the left end of the toolbar
- **ui-mini**: This informs the framework to render a smaller size button
- **ui-btn-b**: This informs the framework to render a swatch b styled button
- **ui-btn-icon-left**: This informs the framework that the icon that is to be used will be aligned to the left-hand side of the button
- **ui-icon-home**: This informs the framework to render a Home icon

Now, we are going to move the current View Current Events button. First thing we need to do is go ahead and delete the current button. Now, add the following code right before the closing of our header div:

```
<a href="#eventregister" data-rel="popup"
data-position-to="window" data-transition="pop"
class="ui-btn-right ui-btn-b ui-btn ui-btn-inline ui-mini
ui-corner-all ui-btn-icon-right
ui-icon-calendar">Events</a>
```

Simple, right? Let's talk about what this code does:

We are creating a link to our main page, index.html (for now) by adding a button that has an icon in it. We have already seen how we style the link tag like a button using the jQuery Mobile specific button classes in the example so far. A few new classes here include:

- **ui-btn-left**: This class informs the framework that the button is to be aligned to the left end of the toolbar
- **ui-mini**: This informs the framework to render a smaller size button
- **ui-btn-b**: This informs the framework to render a swatch b styled button
- **ui-btn-icon-left**: This informs the framework that the icon that is to be used will be aligned to the left-hand side of the button
- **ui-icon-home**: This informs the framework to render a Home icon

Now, we are going to move the current View Current Events button. First thing we need to do is go ahead and delete the current button. Now, add the following code right before the closing of our header div:
Call to Action – Our Main Project

Save the code and you will now see this:

Our mobile app is coming along nicely, isn't it? Now, it's time to analyze the code again:

```html
<a href="#eventregister" data-rel="popup" data-position-to="window" data-transition="pop"
    class="ui-btn-right ui-btn-b ui-btn ui-btn-inline ui-mini
    ui-corner-all ui-btn-icon-right
    ui-icon-calendar">Events</a>
```

Just like our previous button, we are creating a link to our popup and using the calendar icon for our button. The rest of the code is similar to what we used in our original popup declaration and the icon button we just used for the Home link. This brings out the true power of this awesome framework, wherein we can combine different attributes and features of this framework into one single unit to enhance our mobile web application.

We now have a fairly nice header toolbar. In the coming chapters, we will end up changing this header, but it does the job for now. Now, let's create our footer toolbar.

Remove the current contact button we have in our footer and replace it with this:

```html
<a href="#contactpanel" class="ui-btn ui-btn-inline ui-mini
    ui-corner-all ui-btn-b ui-btn-icon-left ui-icon-phone">Contact Us</a>
```
Save this and when you view the project, you should see something like this:

![Image](image.png)

It's quite amazing what a few lines of code can do to a mobile application in jQuery Mobile. Let's finish this section off by looking at that last bit of code we added:

```html
<a href="#contactpanel" data-rel="panel" class="ui-btn ui-btn-inline ui-mini ui-corner-all ui-btn-b ui-btn-icon-left ui-icon-phone">Contact Us</a>
```

We are doing the same thing what we did with the other two buttons. We're creating a link to our panel and assigning the phone icon to it.

That is it for toolbars and here is what our code currently looks like:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Anytown Civic Center</title>
    <meta name="viewport" content="width=device-width, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css" />
    <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</head>
<body>
    <div data-role="page">
```

[81]
Call to Action – Our Main Project

<div data-role="popup" id="eventregister"
data-overlay-theme="b" data-theme="b"
data-dismissible="false">
  <div data-role="header" data-theme="a">
    <h1>Current Events</h1>
  </div>
  <div role="main" class="ui-content">
    <h3 class="ui-title">Sorry, currently there are no scheduled events.</h3>
    <a href="#" class="ui-btn ui-corner-all ui-shadow ui-btn-inline ui-btn-b" data-rel="back">Close</a>
  </div>
</div>

<div data-role="header">
  <a href="index.html" class="ui-btn-left ui-btn ui-btn-inline ui-mini ui-corner-all ui-btn-icon-left ui-icon-home">Home</a>
  <h1>Welcome to Anytown Civic Center!</h1>
  <a href="#eventregister" data-rel="popup" data-position-to="window" data-transition="pop" class="ui-btn-right ui-btn ui-btn-inline ui-mini ui-corner-all ui-btn-icon-right ui-icon-calendar">Events</a>
</div><!-- /header -->

<div data-role="panel" id="contactpanel" data-display="push" data-dismissible="true" data-theme="a">
  <div>
    <p>Contact Us!</p>
    <a href="tel:555-555-5555">(555)555-5555</a>
    <p>1234 First Avenue</p>
    <a href="#eventregister">Anytown, Anystate 12345</a>
    <a href="mailto:contact@anytownciviccenter.com">contact@anytownciviccenter.com</a>
  </div>
</div><!-- /panel -->

<div class="ui-content" role="main">
  <p>Welcome to the website for Anytown Civic Center. We have a lot of great upcoming events for you to check out.</p>
</div>
We used several of the built-in icons for jQuery Mobile in this chapter. There is plenty more. You can find the full list at http://demos.jquerymobile.com/1.4.5/icons/.

Navbars

Navbars is another great widget that is featured in jQuery Mobile. It is a simple navbar but can also be turned into a tab bar that you see in the native mobile applications. It supports up to five buttons and those buttons can also use the icons we used in the last section.

In simple terms, a jQuery Mobile navbar is an unordered list that is wrapped within a div container with data-role of navbar. You can create them in the header or footer of your page.

We are going to add one to our header div element that will serve as our navigation menu in the later chapters. This means for now that it won't be very functional, but the groundwork will be there and we can still see the navbar in action to an extent.

Open up index.html and add the following code before our closing header div element:

```html
<div data-role="navbar">
  <ul>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Facilities</a></li>
    <li><a href="#">Catering</a></li>
  </ul>
</div>
```
Save the file and run the code. You should see the following screenshot:

![Screenshot of a website with a navbar and navigation links.]  

Go ahead and click on the links. The active button will turn blue as you click on it. And now, it's code review time!

```html
<div data-role="navbar">
    <ul>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Facilities</a></li>
        <li><a href="#">Catering</a></li>
    </ul>
</div>
```

We told you that, all that a navbar is; it’s an unordered list wrapped within the navbar data-role container. As you add more elements (remember the limit is five), jQuery Mobile automatically resizes the bar so that the buttons fit comfortably on the screen. If we were to put this code in the footer div element, it would act almost like a native mobile application tab bar.

As previously mentioned, you can use icons in a navbar via the data-icon attribute on the links that we used in the toolbar. Go ahead and practice with that, as well as seeing what this looks like in the footer.

This concludes our code for this chapter. For one last review, in case anything isn't working correctly in your code, here is the full and final code for this chapter. It is important that your code is functioning properly, since this code will form the basis of the rest of the book:

```html
<!DOCTYPE html>
<html>
<head>
    <head>
```
Call to Action – Our Main Project

<div data-role="panel" id="contactpanel"
data-display="push" data-dismissible="true"
data-theme="a">
  <div>
    <p>Contact Us!</p>
    <a href="tel:555-555-5555">(555)555-5555</a>
    <p>1234 First Avenue</p>
    <p>Anytown, Anystate 12345</p>
    <a href="mailto:contact@anytownciviccenter.com">
      contact@anytownciviccenter.com
    </a>
  </div>
</div>

<div class="ui-content role="main">
  <p>Welcome to the website for Anytown Civic Center. We have a lot of great upcoming events for you to check out.</p>
</div>

<div data-role="footer">
  <h2>
    <a href="#contactpanel" data-rel="panel" class="ui-btn ui-btn-inline ui-mini ui-corner-all ui-btn-b ui-btn-icon-left ui-icon-phone">Contact Us</a>
  </h2>
</div>
Summary
This chapter was very code intensive, but at the same time, the code was simple, yet elegant and powerful. This is one of the great things about jQuery Mobile. It allows us to make beautiful and functional mobile web applications with just a few lines of code.

Let's review the topics we covered in this chapter. This chapter focused on some of the most popular and important widgets of jQuery Mobile. We saw that the page widget is the main widget required for the rest of the widgets. We saw how panels can be used to display information and how the popup can be powerful enough to hold an image or form, or serve as a full-blown modal dialog. We rounded off the chapter by seeing how a toolbar can add some nice functionality to our app, and then how the navbar will serve as a nice menu system for later chapters.

Remember to check your code to make sure it is working. We know that we have said this a great deal, but this chapter provides the framework for the rest of the book. If your code isn't functioning properly now, you will have some difficulty in the later parts of the book.

In the next chapter, we will start putting together a navigation system for our application as we link the pages together. We will also explore the touch interactions available in the framework to swipe through the pages we created in this chapter.
Where to buy this book

You can buy Mastering jQuery Mobile from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.