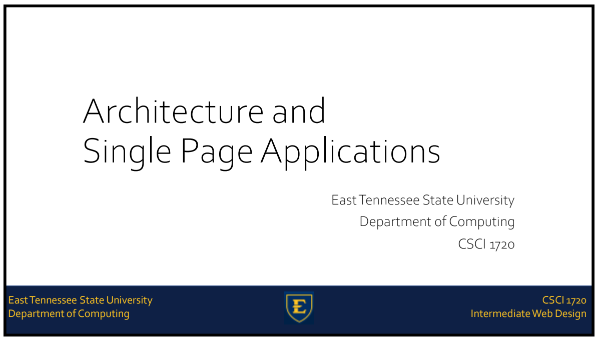
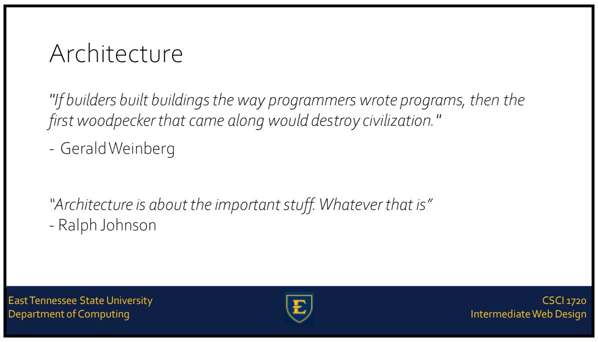


Architecture and Single Page Applications

East Tennessee State University
Department of Computing
CSCI 1720

1



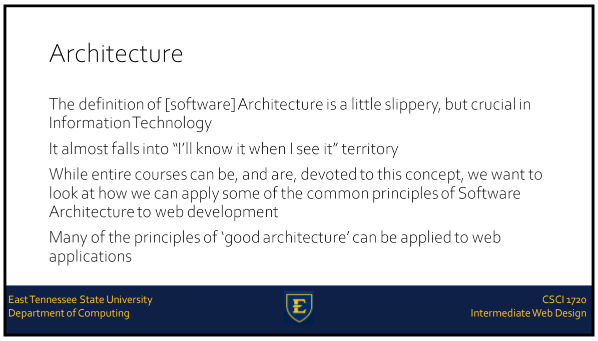
Architecture

"If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization."

- Gerald Weinberg

"Architecture is about the important stuff. Whatever that is"
- Ralph Johnson

2



Architecture

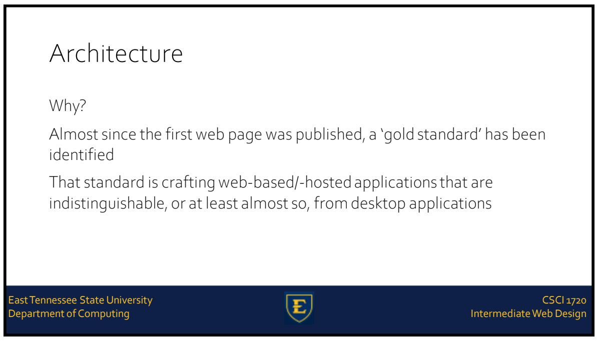
The definition of [software] Architecture is a little slippery, but crucial in Information Technology

It almost falls into "I'll know it when I see it" territory

While entire courses can be, and are, devoted to this concept, we want to look at how we can apply some of the common principles of Software Architecture to web development

Many of the principles of 'good architecture' can be applied to web applications

3



Architecture

Why?

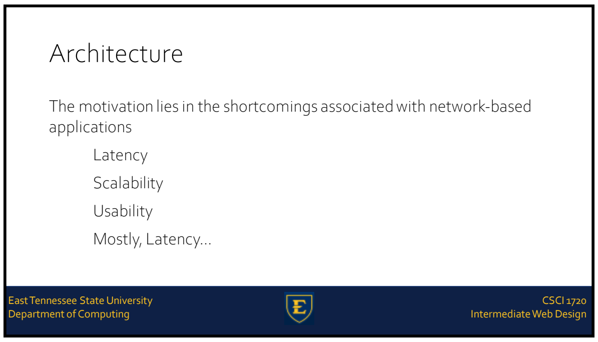
Almost since the first web page was published, a 'gold standard' has been identified

That standard is crafting web-based/-hosted applications that are indistinguishable, or at least almost so, from desktop applications

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

4



Architecture

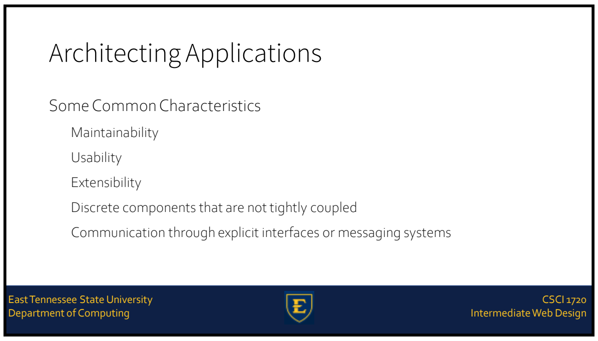
The motivation lies in the shortcomings associated with network-based applications

- Latency
- Scalability
- Usability
- Mostly, Latency…

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

5



Architecting Applications

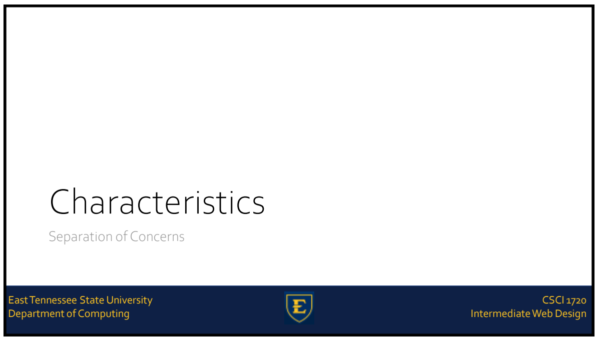
Some Common Characteristics

- Maintainability
- Usability
- Extensibility
- Discrete components that are not tightly coupled
- Communication through explicit interfaces or messaging systems

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

6



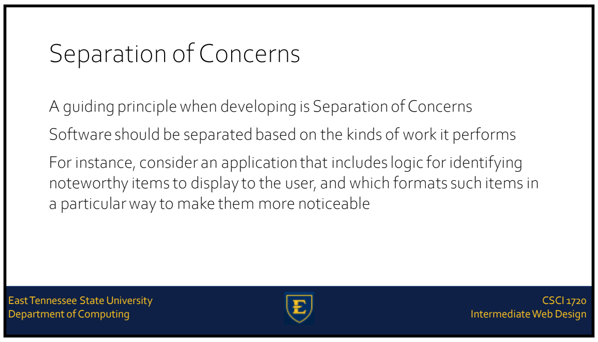
Characteristics

Separation of Concerns

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

7



Separation of Concerns

A guiding principle when developing is Separation of Concerns

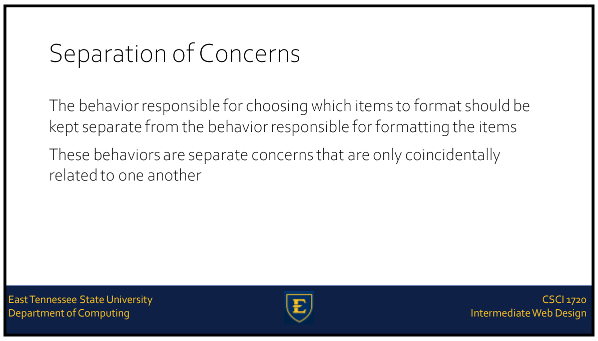
Software should be separated based on the kinds of work it performs

For instance, consider an application that includes logic for identifying noteworthy items to display to the user, and which formats such items in a particular way to make them more noticeable

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

8



Separation of Concerns

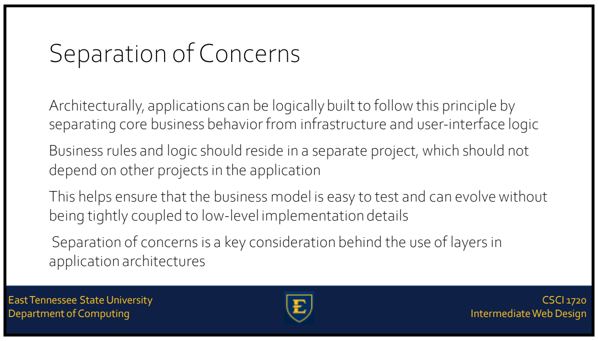
The behavior responsible for choosing which items to format should be kept separate from the behavior responsible for formatting the items

These behaviors are separate concerns that are only coincidentally related to one another

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

9



Separation of Concerns

Architecturally, applications can be logically built to follow this principle by separating core business behavior from infrastructure and user-interface logic

Business rules and logic should reside in a separate project, which should not depend on other projects in the application

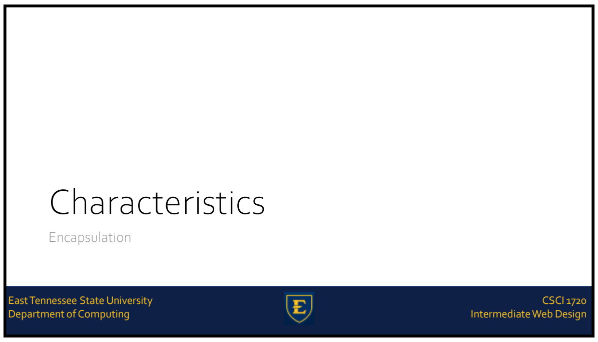
This helps ensure that the business model is easy to test and can evolve without being tightly coupled to low-level implementation details

Separation of concerns is a key consideration behind the use of layers in application architectures

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

10



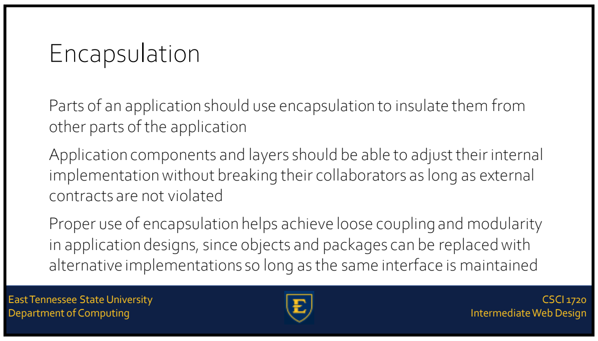
Characteristics

Encapsulation

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

11



Encapsulation

Parts of an application should use encapsulation to insulate them from other parts of the application

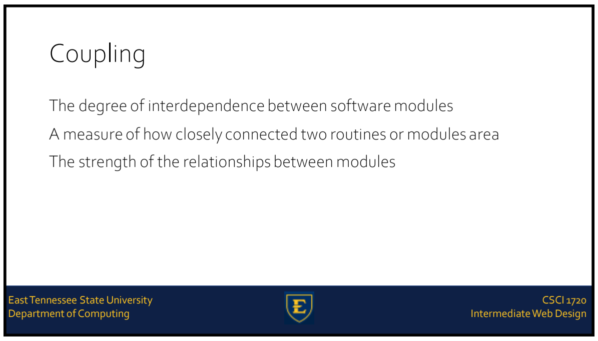
Application components and layers should be able to adjust their internal implementation without breaking their collaborators as long as external contracts are not violated

Proper use of encapsulation helps achieve loose coupling and modularity in application designs, since objects and packages can be replaced with alternative implementations so long as the same interface is maintained

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

12



Coupling

The degree of interdependence between software modules

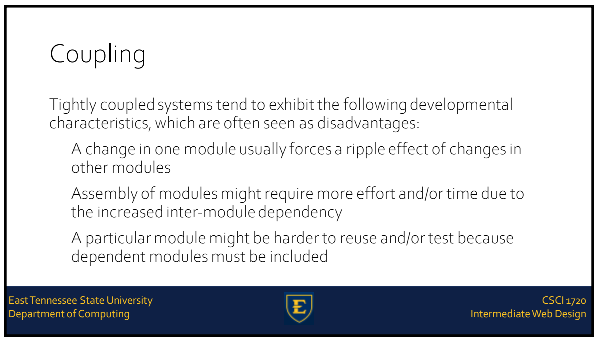
A measure of how closely connected two routines or modules area

The strength of the relationships between modules

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

13



Coupling

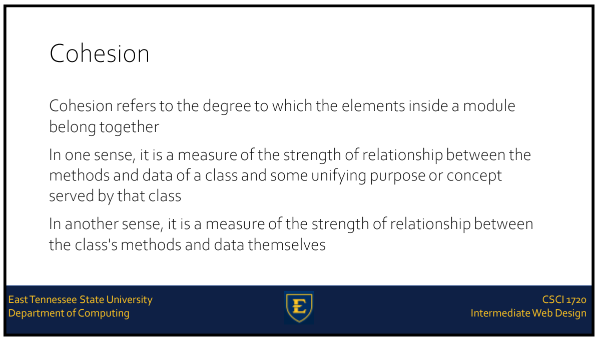
Tightly coupled systems tend to exhibit the following developmental characteristics, which are often seen as disadvantages:

- A change in one module usually forces a ripple effect of changes in other modules
- Assembly of modules might require more effort and/or time due to the increased inter-module dependency
- A particular module might be harder to reuse and/or test because dependent modules must be included

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

14



Cohesion

Cohesion refers to the degree to which the elements inside a module belong together

In one sense, it is a measure of the strength of relationship between the methods and data of a class and some unifying purpose or concept served by that class

In another sense, it is a measure of the strength of relationship between the class's methods and data themselves

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

15

Cohesion

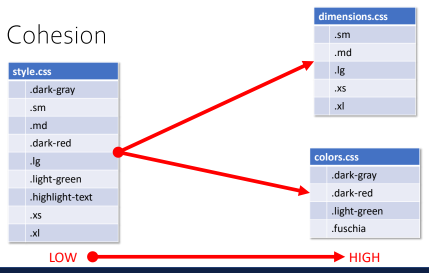
Cohesion is an ordinal type of measurement and is usually described as "high cohesion" or "low cohesion"

Modules with high cohesion tend to be preferable, because high cohesion is associated with several desirable traits of software including robustness, reliability, reusability, and understandability

In contrast, low cohesion is associated with undesirable traits such as being difficult to maintain, test, reuse, or even understand

16

Cohesion



17

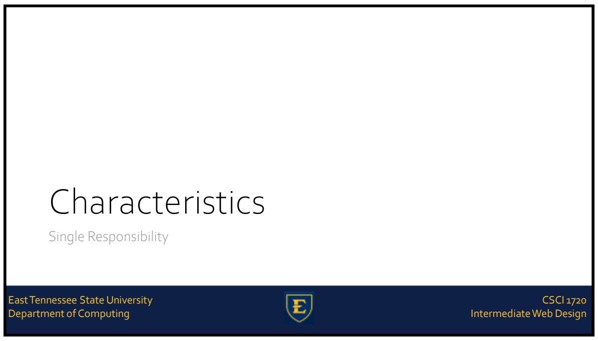
Coupling vs. Cohesion

Coupling and cohesion are terms which occur together very frequently

Coupling refers to the interdependencies between modules, while cohesion describes how related the functions within a single module are

Low cohesion implies that a given module performs tasks which are not very related to each other and hence can create problems as the module becomes large

18



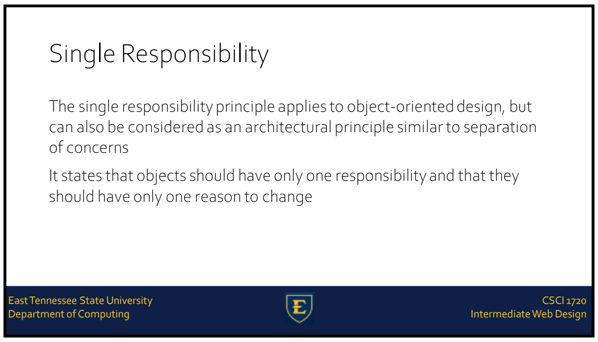
Characteristics

Single Responsibility

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

19



Single Responsibility

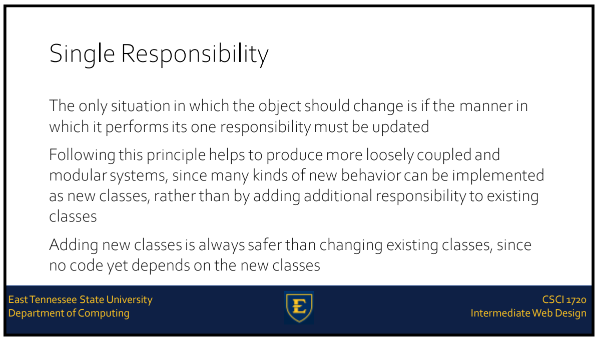
The single responsibility principle applies to object-oriented design, but can also be considered as an architectural principle similar to separation of concerns

It states that objects should have only one responsibility and that they should have only one reason to change

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

20



Single Responsibility

The only situation in which the object should change is if the manner in which it performs its one responsibility must be updated

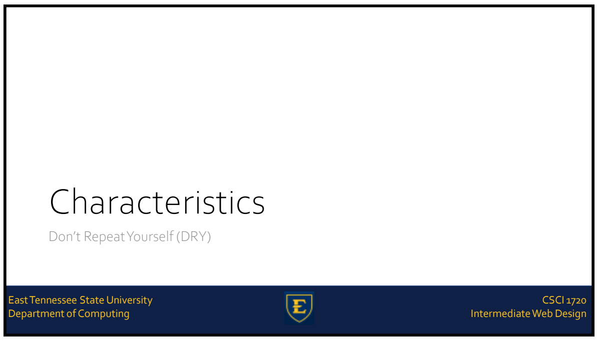
Following this principle helps to produce more loosely coupled and modular systems, since many kinds of new behavior can be implemented as new classes, rather than by adding additional responsibility to existing classes

Adding new classes is always safer than changing existing classes, since no code yet depends on the new classes

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

21



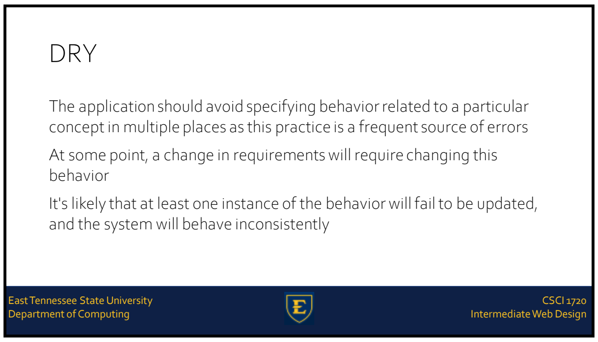
Characteristics

Don't Repeat Yourself (DRY)

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

22



DRY

The application should avoid specifying behavior related to a particular concept in multiple places as this practice is a frequent source of errors

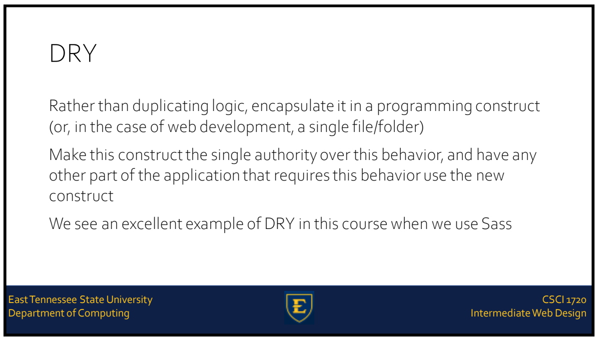
At some point, a change in requirements will require changing this behavior

It's likely that at least one instance of the behavior will fail to be updated, and the system will behave inconsistently

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

23



DRY

Rather than duplicating logic, encapsulate it in a programming construct (or, in the case of web development, a single file/folder)

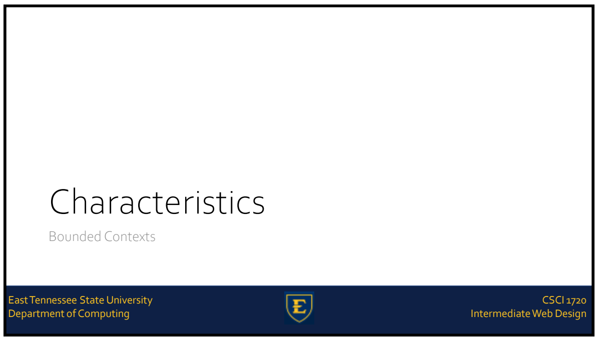
Make this construct the single authority over this behavior, and have any other part of the application that requires this behavior use the new construct

We see an excellent example of DRY in this course when we use Sass

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

24



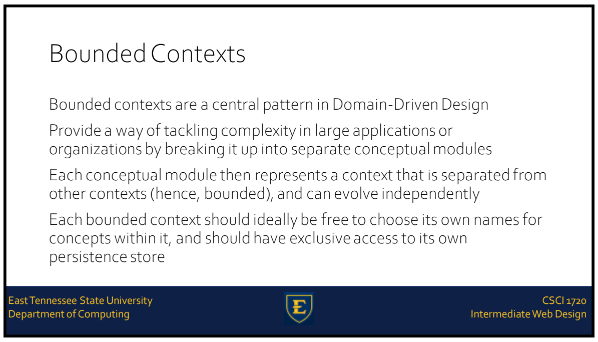
Characteristics

Bounded Contexts

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

25



Bounded Contexts

Bounded contexts are a central pattern in Domain-Driven Design

Provide a way of tackling complexity in large applications or organizations by breaking it up into separate conceptual modules

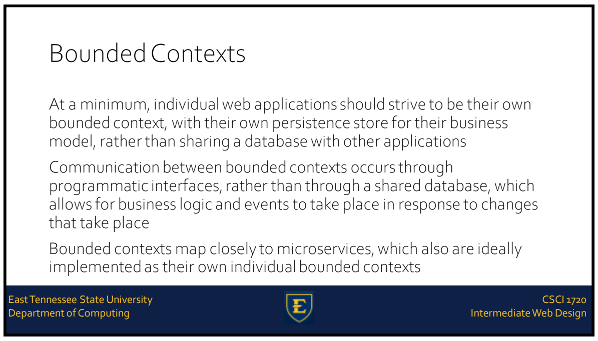
Each conceptual module then represents a context that is separated from other contexts (hence, bounded), and can evolve independently

Each bounded context should ideally be free to choose its own names for concepts within it, and should have exclusive access to its own persistence store

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

26



Bounded Contexts

At a minimum, individual web applications should strive to be their own bounded context, with their own persistence store for their business model, rather than sharing a database with other applications

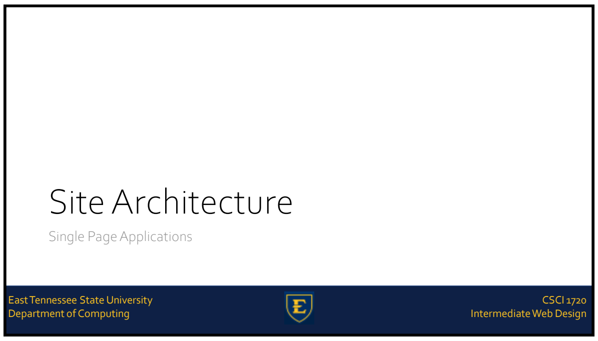
Communication between bounded contexts occurs through programmatic interfaces, rather than through a shared database, which allows for business logic and events to take place in response to changes that take place

Bounded contexts map closely to microservices, which also are ideally implemented as their own individual bounded contexts

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

27



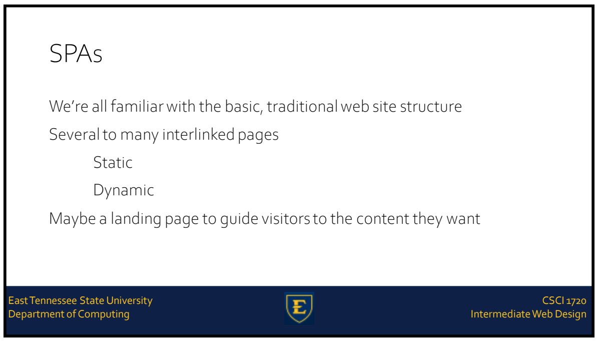
Site Architecture

Single Page Applications

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

28



SPAs

We're all familiar with the basic, traditional web site structure

Several to many interlinked pages

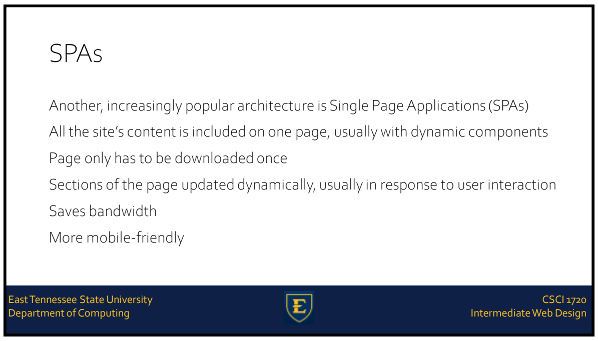
- Static
- Dynamic

Maybe a landing page to guide visitors to the content they want

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

29



SPAs

Another, increasingly popular architecture is Single Page Applications (SPAs)

All the site's content is included on one page, usually with dynamic components

Page only has to be downloaded once

Sections of the page updated dynamically, usually in response to user interaction

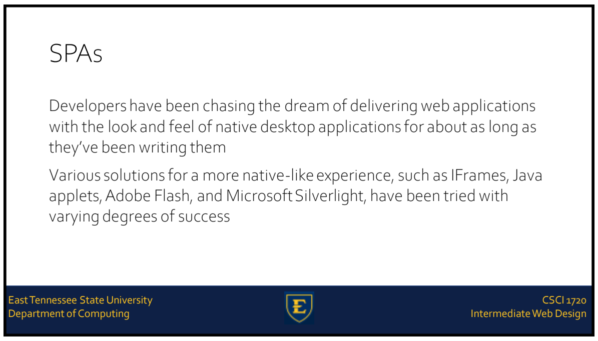
Saves bandwidth

More mobile-friendly

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

30

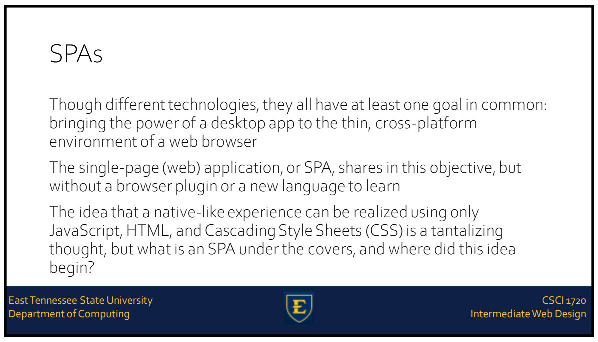


SPAs

Developers have been chasing the dream of delivering web applications with the look and feel of native desktop applications for about as long as they've been writing them

Various solutions for a more native-like experience, such as IFrames, Java applets, Adobe Flash, and Microsoft Silverlight, have been tried with varying degrees of success

31



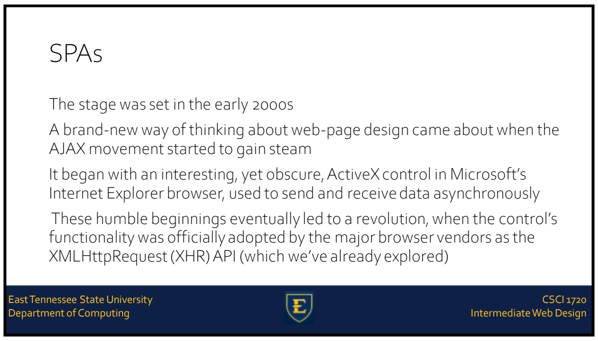
SPAs

Though different technologies, they all have at least one goal in common: bringing the power of a desktop app to the thin, cross-platform environment of a web browser

The single-page (web) application, or SPA, shares in this objective, but without a browser plugin or a new language to learn

The idea that a native-like experience can be realized using only JavaScript, HTML, and Cascading Style Sheets (CSS) is a tantalizing thought, but what is an SPA under the covers, and where did this idea begin?

32



SPAs

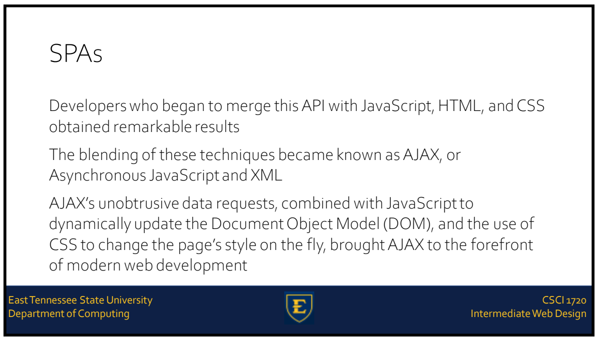
The stage was set in the early 2000s

A brand-new way of thinking about web-page design came about when the AJAX movement started to gain steam

It began with an interesting, yet obscure, ActiveX control in Microsoft's Internet Explorer browser, used to send and receive data asynchronously

These humble beginnings eventually led to a revolution, when the control's functionality was officially adopted by the major browser vendors as the XMLHttpRequest (XHR) API (which we've already explored)

33



SPAs

Developers who began to merge this API with JavaScript, HTML, and CSS obtained remarkable results

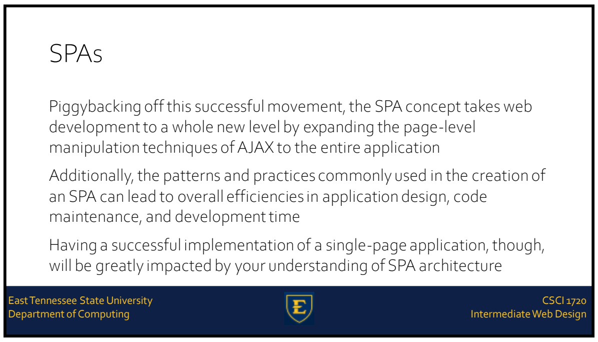
The blending of these techniques became known as AJAX, or Asynchronous JavaScript and XML

AJAX's unobtrusive data requests, combined with JavaScript to dynamically update the Document Object Model (DOM), and the use of CSS to change the page's style on the fly, brought AJAX to the forefront of modern web development

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

34



SPAs

Piggybacking off this successful movement, the SPA concept takes web development to a whole new level by expanding the page-level manipulation techniques of AJAX to the entire application

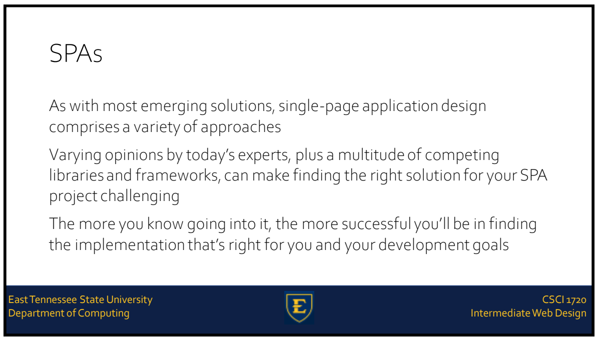
Additionally, the patterns and practices commonly used in the creation of an SPA can lead to overall efficiencies in application design, code maintenance, and development time

Having a successful implementation of a single-page application, though, will be greatly impacted by your understanding of SPA architecture

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

35



SPAs

As with most emerging solutions, single-page application design comprises a variety of approaches

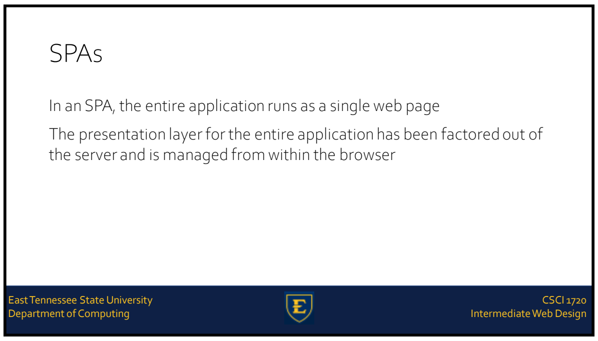
Varying opinions by today's experts, plus a multitude of competing libraries and frameworks, can make finding the right solution for your SPA project challenging

The more you know going into it, the more successful you'll be in finding the implementation that's right for you and your development goals

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

36

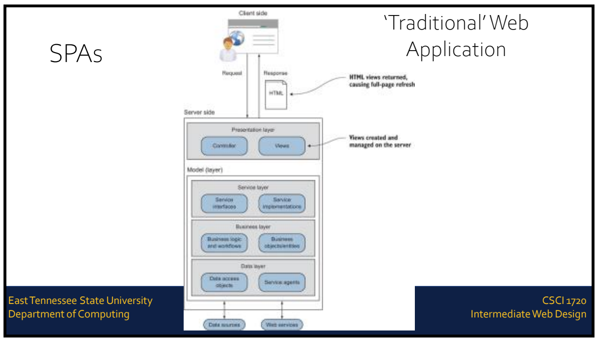


SPAs

In an SPA, the entire application runs as a single web page

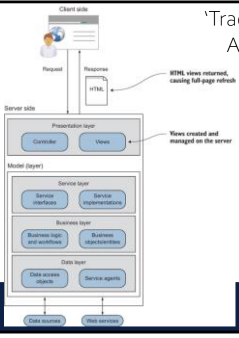
The presentation layer for the entire application has been factored out of the server and is managed from within the browser

37

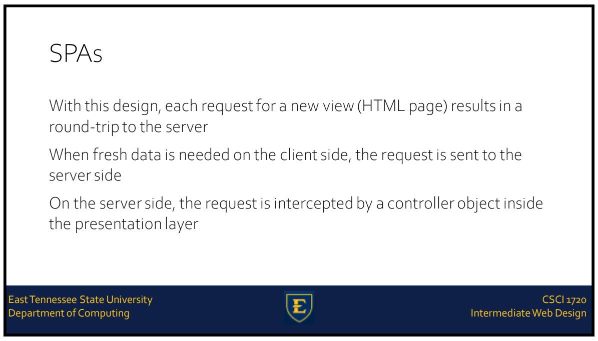


SPAs

'Traditional' Web Application



38



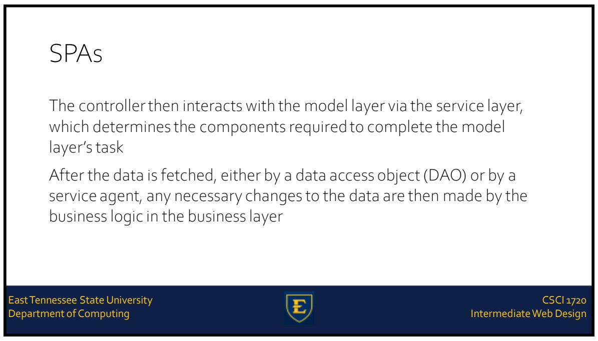
SPAs

With this design, each request for a new view (HTML page) results in a round-trip to the server

When fresh data is needed on the client side, the request is sent to the server side

On the server side, the request is intercepted by a controller object inside the presentation layer

39

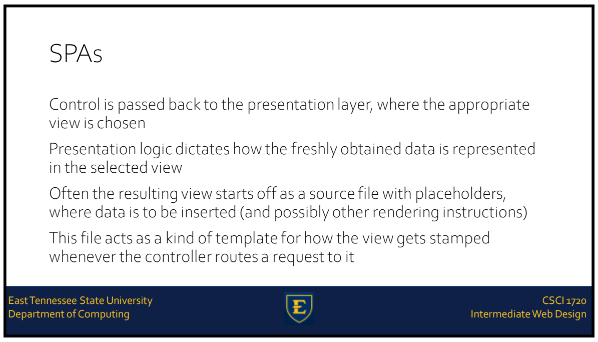


SPAs

The controller then interacts with the model layer via the service layer, which determines the components required to complete the model layer's task

After the data is fetched, either by a data access object (DAO) or by a service agent, any necessary changes to the data are then made by the business logic in the business layer

40



SPAs

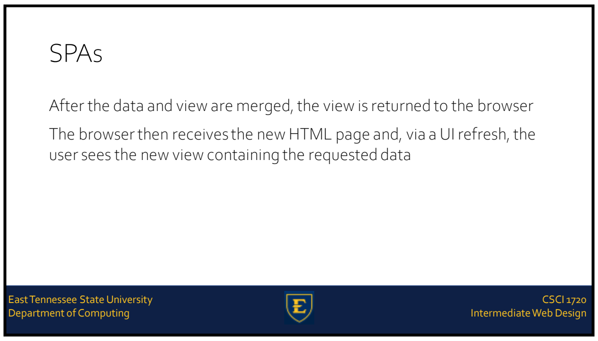
Control is passed back to the presentation layer, where the appropriate view is chosen

Presentation logic dictates how the freshly obtained data is represented in the selected view

Often the resulting view starts off as a source file with placeholders, where data is to be inserted (and possibly other rendering instructions)

This file acts as a kind of template for how the view gets stamped whenever the controller routes a request to it

41



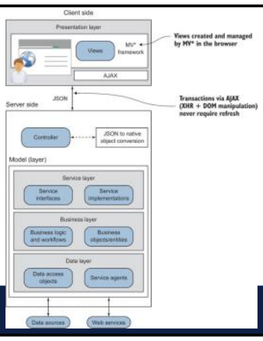
SPAs

After the data and view are merged, the view is returned to the browser

The browser then receives the new HTML page and, via a UI refresh, the user sees the new view containing the requested data

42

SPAs



East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

43

SPAs

Moving the process for creating and managing views into the UI decouples it from the server

From an architectural standpoint, this gives the SPA an interesting advantage

Unless you're doing partial rendering on the server, the server is no longer required to be involved in how the data is presented

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

44

SPAs

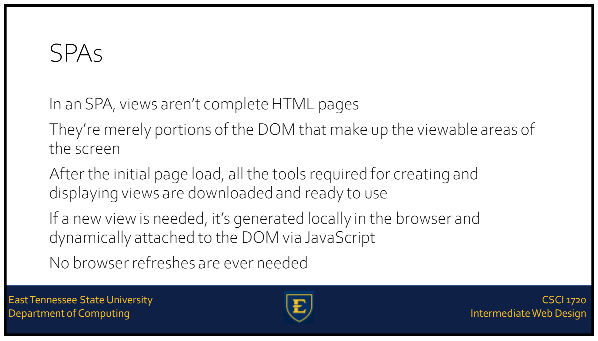
The overall SPA design is nearly the same as the traditional design

The key changes are as follows: no full browser refreshes, the presentation logic resides in the client, and server transactions can be data-only, depending on your preference for data rendering

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

45



SPAs

In an SPA, views aren't complete HTML pages

They're merely portions of the DOM that make up the viewable areas of the screen

After the initial page load, all the tools required for creating and displaying views are downloaded and ready to use

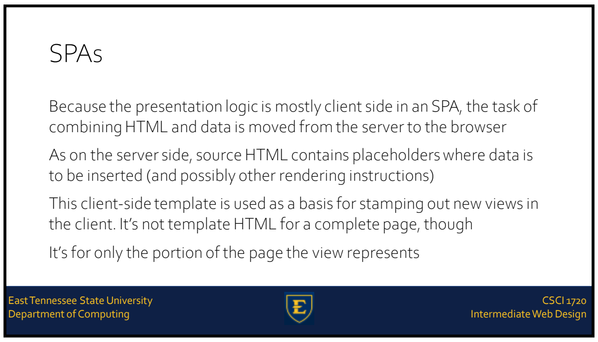
If a new view is needed, it's generated locally in the browser and dynamically attached to the DOM via JavaScript

No browser refreshes are ever needed

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

46



SPAs

Because the presentation logic is mostly client side in an SPA, the task of combining HTML and data is moved from the server to the browser

As on the server side, source HTML contains placeholders where data is to be inserted (and possibly other rendering instructions)

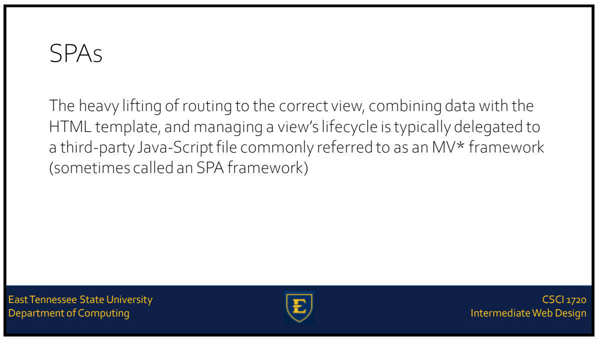
This client-side template is used as a basis for stamping out new views in the client. It's not template HTML for a complete page, though

It's for only the portion of the page the view represents

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

47



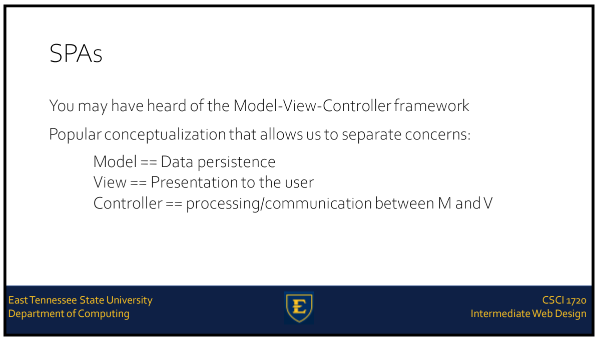
SPAs

The heavy lifting of routing to the correct view, combining data with the HTML template, and managing a view's lifecycle is typically delegated to a third-party Java-Script file commonly referred to as an MV* framework (sometimes called an SPA framework)

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

48



SPAs

You may have heard of the Model-View-Controller framework

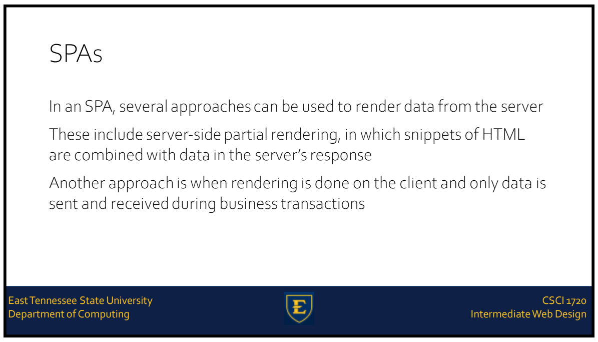
Popular conceptualization that allows us to separate concerns:

- Model == Data persistence
- View == Presentation to the user
- Controller == processing/communication between M and V

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

49



SPAs

In an SPA, several approaches can be used to render data from the server

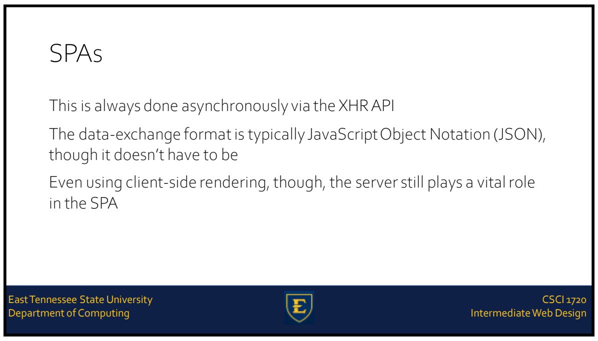
These include server-side partial rendering, in which snippets of HTML are combined with data in the server's response

Another approach is when rendering is done on the client and only data is sent and received during business transactions

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

50



SPAs

This is always done asynchronously via the XHR API

The data-exchange format is typically JavaScript Object Notation (JSON), though it doesn't have to be

Even using client-side rendering, though, the server still plays a vital role in the SPA

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

51

SPAs – The Shell

The single-page part of the SPA refers to the initial HTML file, or shell

This single HTML file is loaded once and only once, and it serves as the starting point for the rest of the application

This is the only full browser load that happens in an SPA

Subsequent portions of the application are loaded dynamically and independently of the shell, without a full-page reload, giving the user the perception that the page has changed

52

SPAs – The Shell

Title
Shell starts empty

53

SPAs – The Shell

Typically, the shell is minimal in structure and often contains a single, empty DIV tag that will house the rest of the application's content

You can think of this shell HTML file as the mother ship and the initial container DIV as the docking bay

54

SPAs – The Shell

```
<!DOCTYPE html>
<html>
<head>
     <title>Shell Example</title>
     <link rel="stylesheet"
          type="text/css"
          href="app/css/default.css">
</head>
<body>
     <div id="container"></div>
</body>
</html>
```

Load the application's style sheets

Initial container DIV

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

55

SPAs – The Shell

The initial container DIV can have child containers beneath it if the application's viewable area is divided into subsections

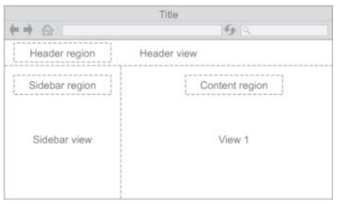
The child containers are often referred to as regions, because they're used to visually divide the screen into logical zones

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

56

SPAs – The Shell



East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

57

SPAs

Regions help you divide the viewable area into manageable chunks of content

The region container DIV is where you tell the MV* framework to insert dynamic content

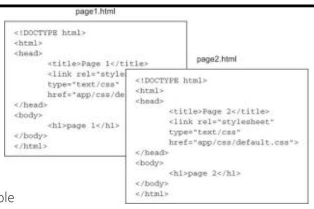
It's worth noting, though, that there are other paradigms

React, for example, uses DOM patching rather than the replacement of particular regions

58

SPAs

Imagining the difference between the average web page and the view of an SPA can be difficult



```
<!DOCTYPE html>
<html>
<head>
        <title>Page 1</title>
        <link rel="style
        type="text/css"
        href="app/css/de
</head>
<body>
        <h1>page 1</h1>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
        <title>Page 2</title>
        <link rel="stylesheet"
        type="text/css"
        href="app/css/default.css">
</head>
<body>
        <h1>page 2</h1>
</body>
</html>
```

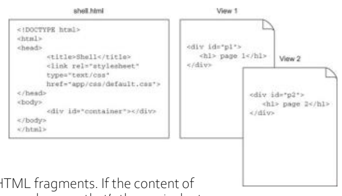
This shows a simple website composed of two web pages

As you can see, both web pages of the traditional site contain the complete HTML structure, including the HEAD and BODY tags

59

SPAs

This shows the same website as an SPA



```
<!DOCTYPE html>
<html>
<head>
        <title>Shell</title>
        <link rel="stylesheet"
        type="text/css"
        href="app/css/default.css">
</head>
<body>
        <div id="container"></div>
</body>
</html>
```

```
<div id="p1">
    <h1> page 1</h1>
</div>
```

```
<div id="p2">
    <h1> page 2</h1>
</div>
```

The SPA "pages" are only HTML fragments. If the content of the viewable area of the screen changes, that's the equivalent of changing pages in a traditional website

60

SPAs

There are many frameworks that implement single page applications

There is a front-end cost – the learning curve

But they streamline the design and development of SPAs, once you've learned them

Angular, React, Laravel, etc.

61

In Practice

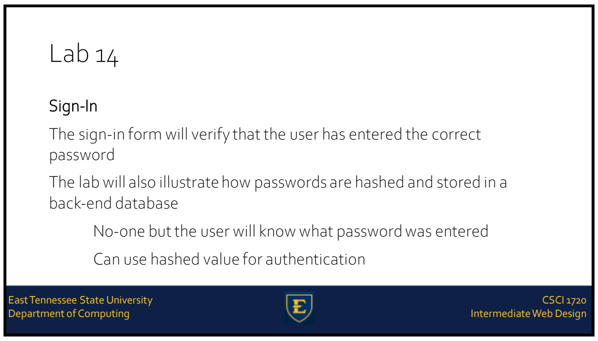
62

Lab 14

This week's lab will illustrate the implementation of a simple SPA – a sign-in/registration page

The page only loads once, but will display a sign-in form, a registration form, or an admin view (a table of registered users) in response to the user's selection

63



Lab 14

Sign-In

The sign-in form will verify that the user has entered the correct password

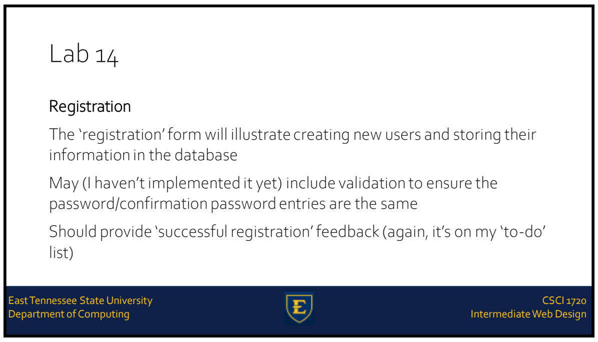
The lab will also illustrate how passwords are hashed and stored in a back-end database

- No-one but the user will know what password was entered
- Can use hashed value for authentication

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

64



Lab 14

Registration

The 'registration' form will illustrate creating new users and storing their information in the database

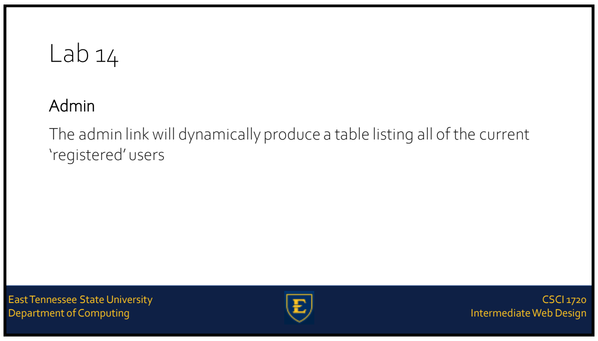
May (I haven't implemented it yet) include validation to ensure the password/confirmation password entries are the same

Should provide 'successful registration' feedback (again, it's on my 'to-do' list)

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

65



Lab 14

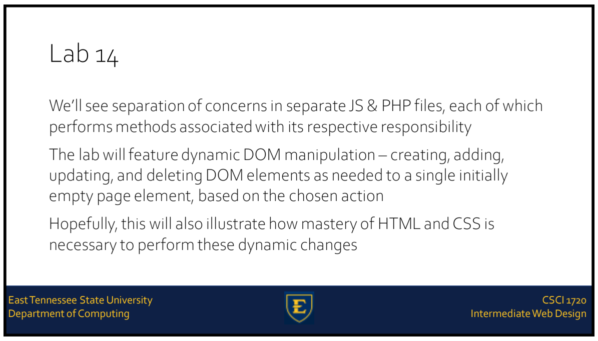
Admin

The admin link will dynamically produce a table listing all of the current 'registered' users

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

66



Lab 14

We'll see separation of concerns in separate JS & PHP files, each of which performs methods associated with its respective responsibility

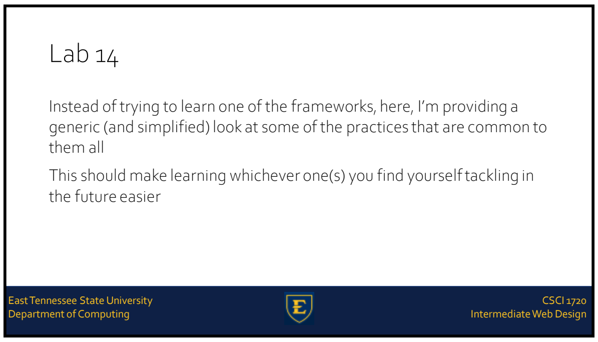
The lab will feature dynamic DOM manipulation – creating, adding, updating, and deleting DOM elements as needed to a single initially empty page element, based on the chosen action

Hopefully, this will also illustrate how mastery of HTML and CSS is necessary to perform these dynamic changes

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

67



Lab 14

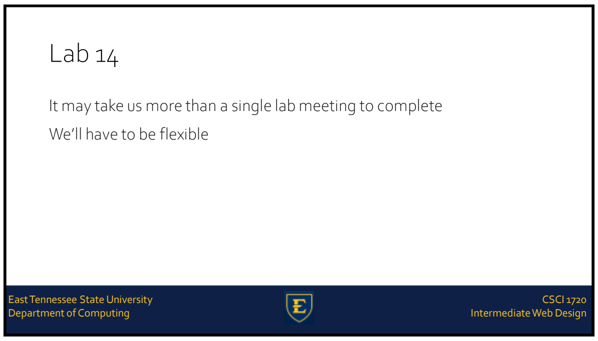
Instead of trying to learn one of the frameworks, here, I'm providing a generic (and simplified) look at some of the practices that are common to them all

This should make learning whichever one(s) you find yourself tackling in the future easier

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

68



Lab 14

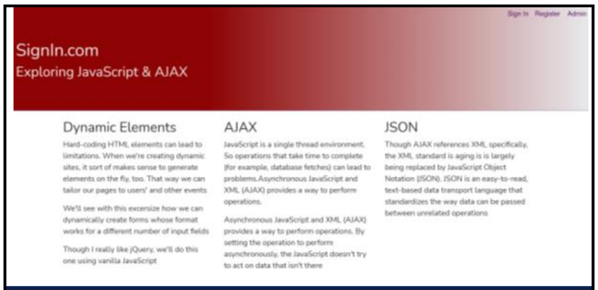
It may take us more than a single lab meeting to complete

We'll have to be flexible

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

69



Sign In Register Admin

SignIn.com
Exploring JavaScript & AJAX

Dynamic Elements

Hard-coding HTML elements can lead to limitations. When we're creating dynamic sites, it sort of makes sense to generate elements on the fly, too. That way we can tailor our pages to users' and other events

We'll see with this excersize how we can dynamically create forms whose format works for a different number of input fields

Though I really like jQuery, we'll do this one using vanilla JavaScript

AJAX

JavaScript is a single thread environment. So operations that take time to complete (for example, database fetches) can lead to problems.Asynchronous JavaScript and XML (AJAX) provides a way to perform operations.

Asynchronous JavaScript and XML (AJAX) provides a way to perform operations. By setting the operation to perform asynchronously, the JavaScript doesn't try to act on data that isn't there

JSON

Though AJAX references XML specifically, the XML standard is aging is is largely being replaced by JavaScript Object Notation (JSON). JSON is an easy-to-read, text-based data transport language that standardizes the way data can be passed between unrelated operations

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

70

Sources

https://martinfowler.com/architecture/
https://livebook.manning.com/book/spa-design-and-architecture/

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

71