**Ting’s Web Page**

Sang Tran

Professor: Dr. Zhang Ting

University of Houston Downtown, Houston, Texas 77002

# **Abstract**

The online teaching staff directory of the University of Houston Downtown (UHD) stores an extensive list of information regarding all UHD’s teaching staff, their contact information, and other important information such as publications and degrees. However, there are much to be desired. The ability to customize the style and information displayed on the staff page of each member is extremely lacking. The main goal was to create an extensive website template for the presentation of each staff member’s information as an innovative and customizable website.

The main programming languages used are HTML, CSS, and JS. The template incorporates a dynamic JS based navigation that makes it easy to add and remove new pages without having to modify all existing pages. Each page (biography, projects, teaching experiences, …) has its own vibrant style and lively animation. With the well-coordinated efforts of every team member, the website template incorporating Dr. Zhang Ting’s personal information page had been completed successfully.

The finished template included a plethora of different customizable styles for each page. Each page’s main container has a convenient built-in css fade in animation, pages such as research experience and activities also have more eye-catching gradient scrolling animation on mouse hover, giving the page a much more unique and vibrant look.

# **Introduction**

With intense competition between each university for enrollment in each semester, it is imperative that the information regarding the university’s teaching staff be updated to be more engaging for potential students to read. While the UHD’s staff directory do properly list the information for each staff member, the lack of customization and vibrant style does little to encourage new students to enroll. There is a need to create a new vibrant and engaging website template for each staff to be able to use and customize to their heart’s content. The final goal is to provide the template for each staff member to use and create an amazing personal information site so that students who are looking to take their classes can see all their relevant information in the beautiful and engaging new website template.

Because the template needs to not only be available on all browsers and all platforms, but also easy to modify and customize, we have decided to start work using the popular and easy to use Bootstrap 4.0 framework. The framework is doing most of the heavy duty tasks such as managing detailed animation and the look and feel of the html components. Since each team member wanted to try for their own unique look, each page is modelled after their own preferences, creating a personality for every page.

**HTML and CSS Design**

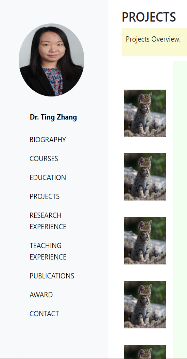
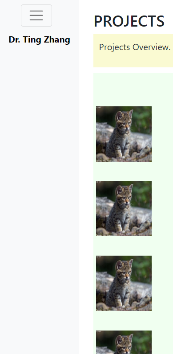
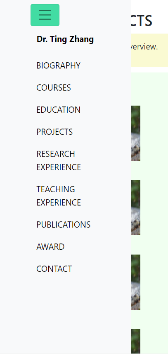
The HTML within the faculty page has separate HTML files for each section, the CSS is contained in one file as well as embedded in HTML. HTML files contain the content of the pages, while the CSS contains the styles that affect the pages appearances. This section covers a majority of the elements used for the faculty page.

**Navigation Bar**

Through the development of the navbar, the implementation started with HTML and CSS. Also, when hovering over a navigation section name, the section name will turn a gray color; by default the text will be colored black in the navigation panel and can be changed within .nav-item a in CSS. The name is aligned center with the name class. The id, nav1, indicates in that the container floats to the left, which allows the navbar to be positioned on top of other items at the left. The navbar items are also vertically aligned and have a max width of 256px. The profile image is made into circle with border-radius at 50%. Navbar also collapses when displayed on smaller screens and expands upon clicking the menu button.

The navigation bar is located on the left and has a profile image, a text field for the person’s name, and the sections. The navigation bar is contained in an unordered list which has the sections linked to a separate page for the respective content. The content of the navigation bar (navbar) are padded to keep a clean appearance and avoid contact with the borders. The image can be changed by changing the image name to an image located in the respective img folder. The elements are centered, with the section names aligned left. The current page that the user is on will contain the active class. The navbar will remain docked to the left as the user scrolls vertically through the page with a fixed position and remains on top of other page content when scrolling horizontally with the z-index (stacking element) set to a positive value.

The HTML displays the navbar as an unordered list with an ID attached which will lead to display of navigation items (handled by JavaScript). The code for the navigation bar used to require editing the navigation bar for each page. However with JavaScript, the HTML code for navigation bar is located in one place. The navbar also displays the professor name in collapsed navbar on smaller screens. The faculty page is suited more for larger screens at the moment, with the current navigation bar.

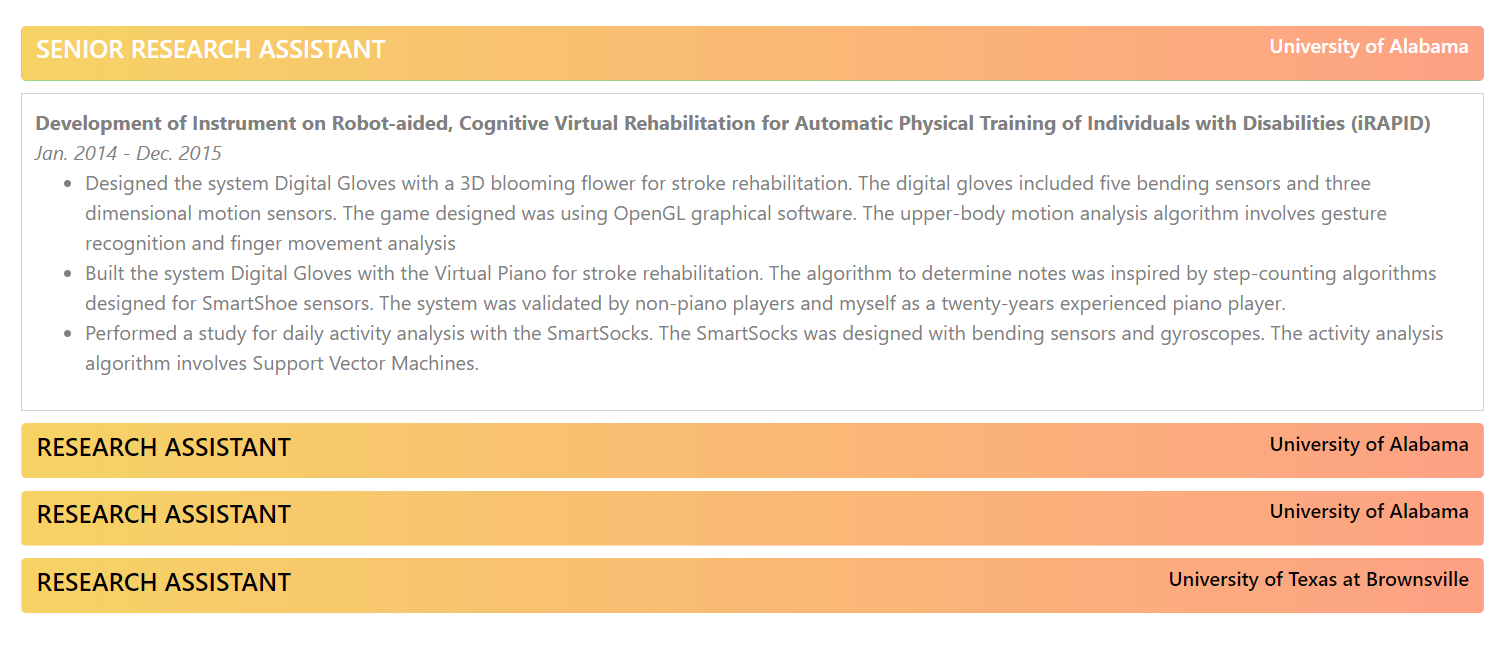
  

Navigation-Bar Menu. Expanded for large screens (left). Collapsed for small screens (center). Expanded for small screens (right).

**Panels**

Some of the panels used are able to expand and collapse to reveal information. Within the Experience page, the collapsible panels are buttons. The buttons will have a data-target which will be displayed when the button is clicked. Also, for further manipulation, the aria-expanded attribute is added to change the appearance of the panels and buttons in the external CSS file. The panels are encased in a panel group, and each panel will have the button and the target panel information. For the Experience page, the collapsible panels are initially false and displays the position text on left and the location text on the right of the panel button, and have padded space around each panel to create a distinction between each panel. The b, i, ul, and li elements are used for the panel information. The overview sentence in the panel is bold; the date is italic; the important details are bulleted.

The panel button will have a green background and white text when the aria-expanded attribute is true, meaning the panel is expanded. The panel will also have a border around the information with padding around the text. A panelText class is attached to the panel text to generate a light gray solid 1px border. Whenever the mouse hovers over the panel buttons, there will be no outline or effects to indicate the hover. Also, because the default bootstrap button success class has a box-shadow, a custom class is attached and removes box-shadow from the panels. The panel group has a block display, so it will be in box container and is positioned relatively, in accordance to the normal flow. There is a transition effect, ease in, that is used to try and smooth the effect of collapsing and expanding the information.



Collapsible and Expandable Panels

The publications page has the id, publications-overview, which wraps the text to right of the navbar and presents a light purple background color with a dark purple border for each section. The width is set to 1200px to fit for larger screens. The publications are unordered lists contained within paragraph elements, separated by headers and a small star character to the left of each header. The bold (or strong) and underline elements are used throughout the publications page.



Publications page with unordered list, bold and underline elements and grow over hover effect

The projects page displays the projects via scrolling and is centered to the container right of the navbar. The project name is displayed at top of each section indicated by a break. The location and date is beneath the project name, followed by an abstract and image(s) of the project. The project sections imitates a Stackedit style. There is also a line to separate the project name and rest of project information. The heading texts have a margin around them.

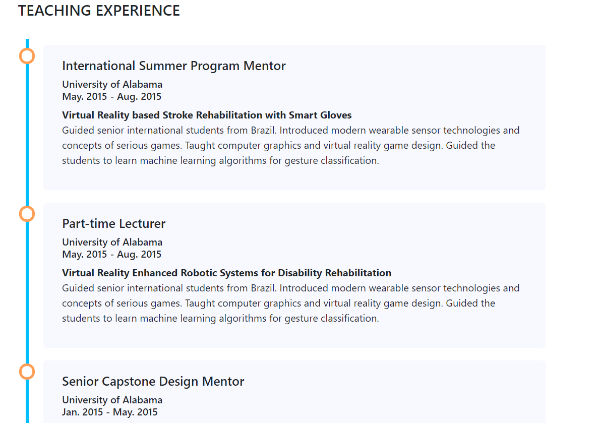


Centered Projects page

**Timeline**

The timeline is used in the teaching experience section with the latest experience on top. The timeline is aligned left and is generated with mainly CSS. Within the HTML file, each timeline event is in a container with container\_t class. Each event also has a background fill color to separate the events from each other and the background. The white and orange circles are added and aligned within the after pseudo-element. The vertical timeline is blue with 6px width.

Each event has padding to contain the text without overflowing. The text will reflow if the screen window is resized. The events have a description paragraph, a bolded text of the position and a topic sentence. The date and location is beneath the position.



Timeline

**Animation and Transition Effects**

A SlideUp animation is made and used within the sample projects page to animate the project description appearing upon clicking an image to the left. An overlay on hover effect is also used to apply a box shadow effect and slightly opaque gray background upon hovering over the image(s). The overlay effect inherits its width and height and scales the gray box by 1.1 to cover the image and add a thin border. A fadeInRight animation is also used on the pages, which will bring in elements from the right with a second duration and ease-out effect. There is also a grow animation on hover over of publications’ panels.

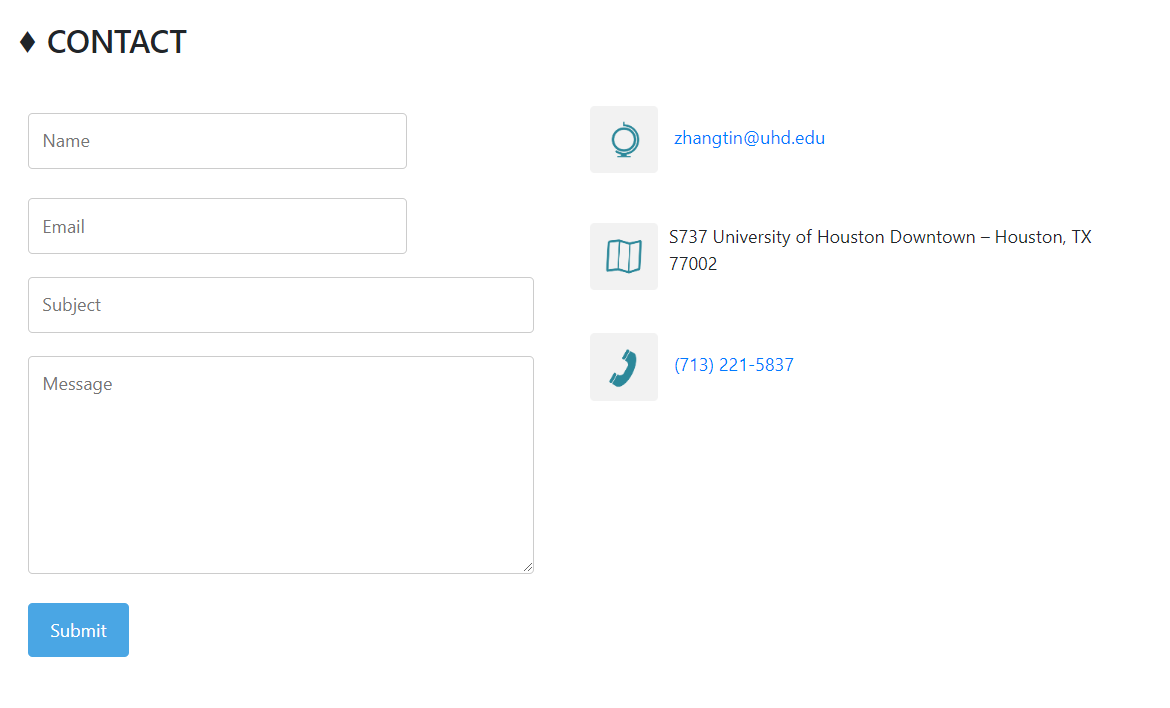
**Padding and Margin and Headers**

The projects page has a max-width apply to 90% to avoid overflow of information from the background container. The publications.html page contains a light goldenrod yellow background color and contains 10px padding around the text. The publications are separated by sub-sections with unordered lists.

The headings for the sections’ individual pages are h3, a moderately large text size. The education. The individual cards will also use headers for main sub-sections’ titles or titles of degrees or positions. The biography page header contains the professor’s name and title in a different color. Then an overview paragraph follows after with professor’s photo on the right. The section headers also have a special character to the left of the heading (e.g. a diamond character) which can be removed or changed if desired.

**Form, Email, and Phone Number**

Within the contact page, users can see a form with name, email, subject, and message input boxes and a submit button. The inputs and textarea (message text box) have margin of 10px to add spacing between the elements. These inputs are contained within the form element. There is currently no form action, so when the submit button is clicked, the form redirects to the current page. Professor’s contact information (email, office location, and phone number) is located to the right of the form. The email and telephone number has href attributes, which will redirect to a mailing application and a calling app (or pick an app if no calling app is detected) respectively. The contact info and form both have the fadeInRight animation applied.

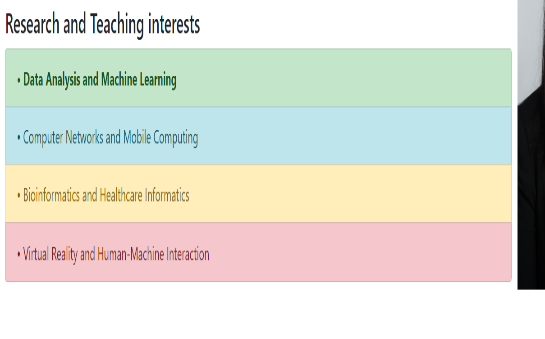
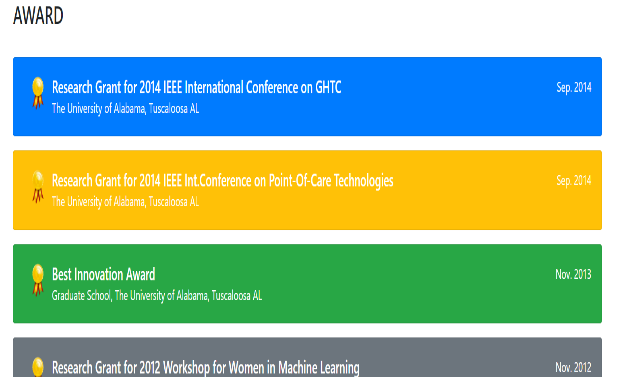
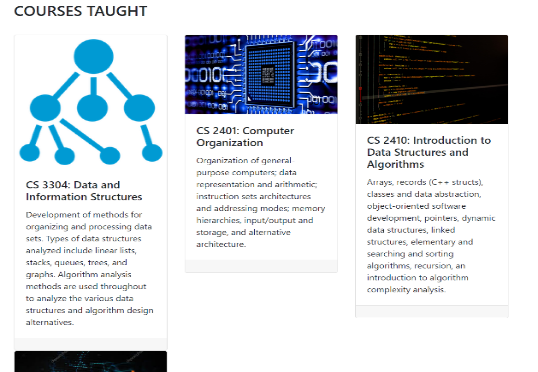


Contact form (left). Contact Info (right).

**Cards**

The cards have a header, image, and information. Majority of the cards use the bootstrap CSS default classes to stylize the cards. Within a card is an image, a title header, and brief overview information. The card text and title is contained in a card body. There is also a card footer which can be updated manually to include a specified time (since last updated); however, is removed for the courses page. The cards are used within courses.html. The card body for the courses cards will have the course description that will be wrapped in the dimensions specified by the container.

The other card type used is in the award.html page which lists the professor’s awards with the date floating to the right and the award name and location floating to the left. The awards cards have various colors. The biography page has cards which contain the professor’s research and teaching interests. The services and activities pages also have panels in card style to separate the different services and activities and the corresponding information padded inside.



Course cards with various widths (left). Award(s) with different background colors (center). Interest cards (right).

**JavaScript Design**

JavaScript design can best be described as front-end functionality. Since JavaScript is an interpreted language and not a compiled language, JavaScript can be executed in the browser. For the JavaScript functionality, we primarily relied on Twitter Bootstrap to provide us with what we needed. This library allows us to have simple user interactions such as tooltips and button pop-ups. We also might need something like a carousel to present images to our viewers, so we can use this as well.

JavaScript can also be used to do things like front-end validation. Why we would want to do something like this is because it provides a better user experience. A user can find out if their password or username works before sending the page to the server. One other application of JavaScript on the front-end is to do simple arithmetic. If we want to know how many hours the professor is teaching, we can add up the integer count of the hours of each individual course.

There has been a recent push to get JavaScript organized into certain design patterns. One of the more popular design patterns is MVC, or model view controller. This allows programmers to design their JavaScript code into logical patterns. The first being a model. The model is basically the data model, or what the database back-end will communicate with using a REST API or something similar. The View layer of the code refers to everything we see as the end user. This includes all user interaction with the application. Finally, the controller layer of JavaScript is the brains of the applications. This performs all the logic operations of the application using JavaScript.

**II. Backend**

The backend, or the database, is used to store persistent data in our web application. Normally, a mainstream database like MySQL or Oracle is used to hold our data. The data is stored into tables that we can use to query the database. The language that we can use to interact with the database is called SQL. SQL is an easy to learn language that we can use to pull data from the database and into the application. We can write queries to both select data out of the database and insert data in the database. Often, we want to be able to do the following; create, read, update, and delete. These are the goals for most web applications.

What we can also do with the SQL language is to create something called stored procedures that run during the lifecycle of the application. For example, if a user wanted to insert data into the application a store procedure executes an insert statement. The Node.js server used prepared SQL statements. So, we can run SQL statements within our application without having to access the command line. SQL commands can also be created to have views into the databases. These views are virtual databases that can be executed when a user wants to access a specific part of the database. Features like the ones described above are why we are able to insert data into the database and retrieve that same data.

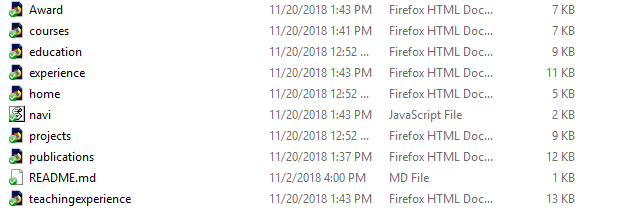
The database cannot directly interact with the front-end html and JavaScript. There must be backend server code that interacts with the database. This can be done using PHP which we are learning in class or JavaScript. JavaScript back ends are called Node.js, which is a JavaScript library that is used to handle http requests and things along the same lines. Node.js is an exciting new library that allows a web program to built all in JavaScript from front end to back end.

**Update and maintenance instructions**

Below is an example of files available on Web Application server. Navigate to appropriate folder to find the below files. For example, the might could possibly be.

//server-uhd/webserver/csfaculty/webpage/etc….

You should see the below files, or similar files within this folder.



Steps Instruction:

|  |  |
| --- | --- |
| 1 | Open file named \*.html in Visual Studio, where \* is the name of file you would like to change. |
| 2 | Find the section of the page you would like to edit by typing control+ F. |
| 3 | Once in the section of page you would like to edit, find the text you would like to edit.  This will be enclosed in html, for example <p> or <div>. DO NOT edit this part.  Instead only edit text in between html. |
| 4 | Carefully type text into this section.  Please do not copy and paste text into edit area as this can cause formatting issues. |
| 5 | If you would like to add similar content.  For example, new paragraph or bullet point.  Copy and paste from within a section in Visual Studio. |
| 6 | If you want to add an image.  Place the image file into the img folder.  The source of the image then would be the name of the file. |
| 7 | To add another page to the website.  Copy and paste an exisitning page and then rename the file to something else.  For example, activities.html would become research.html. |

Demos of changes possible within Visual Studio:

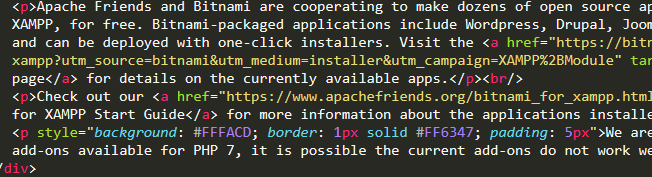


Figure 1. Possible text to be edited

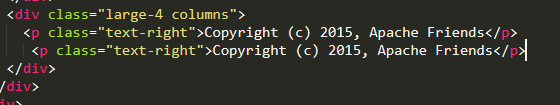


Figure 2. Copy and paste new content

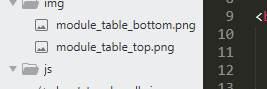


Figure 3. Adding images to image folder

## **Database**

Because the project is a static website that provides information about Dr. Zhang Ting, there are no database system to be implemented.

However, because the template format used was very easy to reuse and distribute to other professors who would like to create their own personal website, it is possible to expand the current template into a detailed database system used to host multiple professors website information.

Possible Entities:

* Biography
  + professorID (PK): the unique ID for each professor
  + Biography: the biography text for that professor
  + Contact\_number: the phone number for their office
  + Email: their email address
* Degrees:
  + ProfessorID (FK): the foreign key pointing to the professor’s ID
  + Degree name
  + Degree\_completion\_date: the date the degree was obtained
  + Institution\_name: the name of the institution that granted the degree
* Classes
  + Class|D (PK): the unique class’s ID
  + CourseID: the id number of the class’s course
  + Room\_number: the string of the room number
  + Time\_start: time value of the class’s start time
  + Time\_end:  time value of the class’s end time
  + Days: can be “M”, “Tue”, “Wed”, “Th”, “F”, “S” or any combination of the two
* Experience
  + Time\_start: time value of the experience/event’s start time
  + Time\_end: time value of the experience/event’s end time
  + Experience\_name: name of the event/experience
  + Experience\_desc: description of the event/experience
* Publications:
  + Date: the publication’s date
  + title: the publication’s title
  + Location: the publication’s location

## **Hosting**

There are many choices when it comes to finding a free website hosting service, below are some of the most suitable choices:

**GitHub Pages**

Github provides free website hosting with a bandwidth limit of 100 mb per month

Instructions:

* Create an account on Github
* Create and publish a repository with the name “username.github.io”
* Upload the website’s files onto the repository, make sure the homepage is named “index.html”

**Using the provided student or staff account**

Each student and staff in UHD is provided with a free linux account on the school’s server. A static website can be hosted there for free

Instructions:

* Using an SSH client like Putty, connect to gator.uhd.edu, port 22
* Enter username and password
* Make a folder named publich\_html
* Use FTP to transfer website’s files into that folder
* The website is now online at gator.uhd.edu/~username

**Host on the school’s servers**

Instructions: contact the IT department and provide them with the website’s files so that they can host them on the school’s own server.

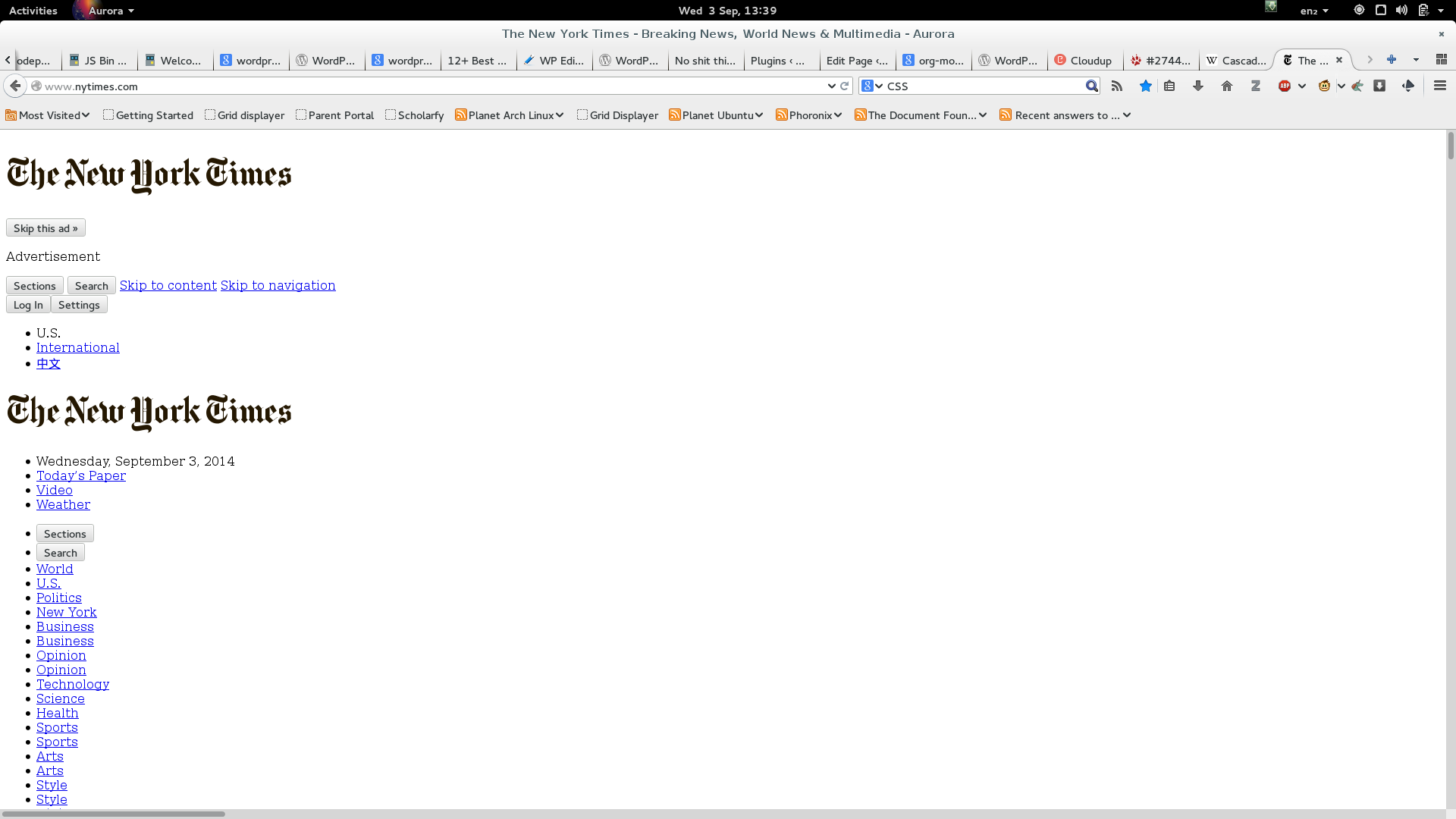
## **Deployment**

After properly hosting the website on a suitable hosting service, the site can be deployed at any time. If needed, contact the IT department to have them direct the school’s url (for example, uhd.edu/professor\_name) to the hosting service’s url.

**Conclusion**

Technology in general is still relatively new, including HTML, CSS and JavaScript, and they are still evolving. As previously stated, these languages are used to design and format web pages. Just over 20 years old, new ways of using markup and server scripting languages are still being discovered. HTML5 is the updated version of HTML, which includes audio, video, and utilizes databases for storing data, and much more. Although we did not use these on our webpage, they are great tools and improvements in web development. The latest version of CSS is CSS3, which has a lot of new upgrades as well. For example, CSS3 includes box model margins, more text effects, and backgrounds and borders. We used many aspects of CSS3 to make our webpage look more appealing and creative. Using a light background with pops of color grabs the reader's attention and pulls them in to the content, without even realizing it.

HTML and CSS are important because together they are used to make webpages and to make them look interesting (see image to right of *The New York Times* website without CSS). This aspect increases user engagement and in turn can be beneficial for businesses in all different fields, such as retail, education, and social media. For our project, it is important for professors have custom webpages in order to draw in students. A creative webpage will persuade students to enroll in courses with these professors, and for students to be educated about their professors’ experience. In addition, these webpages could be utilized to post assignments and could be used as a form of communication with professors. Therefore, if professors have no webpage or a dull webpage, it will leave the students disinterested.

Another important aspect of web design is JavaScript. With JavaScript being used to create interactive effects on websites, this language is very beneficial in increasing user engagement with user friendly objects. It is also used to calculate and manipulate data, based on user input. A recent and popular update to JavaScript is the autofill in on forms, based on user settings and information. This helps the user by not having to type so much information. However, since our webpage is simply being used to display professor’s information, we did not use the language to its’ full potential. However, we did take advantage of its’ ability to display information in a dynamic and fun way. This is important to attract readers to certain parts of a webpage. Without JavaScript, webpages would not be as user friendly or as entertaining.

In addition, databases are very important in web development. Although our webpage did not include backend databases, it is important to acknowledge the importance of them. Backend databases are used to store and manage data such as transactions or websites that depend on registrations. All these aspects are what make the world wide web. It is fascinating to watch web development expand, and it is exciting to think what the next development could be.

**Future Work**

Due to technology constantly evolving and new ideas being discovered, there is always room for improvement in web development. HTML, Cascading Style Sheets and JavaScript could expand in the future and open a door to more creativity for websites. For example, some current web development discoveries that are still improving are artificial intelligence and virtual reality. These ideas are still expanding, but they could be implemented in the future in terms of educational websites. CSS and JavaScript upgrades could add more options to professors’ personal webpages, as well as the university’s website. For example, the professor’s webpage could grow by adding databases to the website. A student log in could be created for each student to access their grades, communicate with each other in a public forum, and view announcements from the professor. Another JavaScript feature that would be interesting to see on the professor personal page would be a weekly poll or quiz section, that could act as an interactive way to test each student’s knowledge. It could simply be used to review material, without the pressure of a grade. This function would have to be user friendly for the professor to update the question(s) with ease.

Another way to expand on the professor’s personal webpage would be to add more graphics, to make it more appealing and interesting. It could also be used for the professor to post video lectures for the students for each specific course. A jeopardy game would be a fun and cool aspect to the website as well. The website could have the function of the game set up, and all the professor or student would need to do is type review questions and answers. It could also be a multiplayer game with other students online. This aspect of the website could serve as a learning tool and review guide for students. It would allow students to have easy access to study material in order to help them in the course. In order to not have all the work rely on the professor, it could be accessible by students to post their own reviews, so everyone could benefit from it. This part of the website could be accessed under the ’Courses Taught’ page (see image).

These are just a few ways the website could expand in the future. Also, it would be remarkable if each professor adopted the idea of a personal website. Moderate changes to the website would be crucial in order to keep it updated and current with modern designs and themes. Overall, the options are limitless to expand on the idea of a personal webpage for professors. However, the payoff would be worth it if would be increase student engagement and student interaction. Also, it serves as a way for professors to have their achievements acknowledged and an accurate display of their experience.

**Reference**

Boostrap 4.0 Documentation:

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

W3school an education website:

<https://www.w3schools.com/howto/howto_css_timeline.asp>

Animated Accordion CSS:

<http://www-db.deis.unibo.it/courses/TW/DOCS/w3schools/howto/howto_js_accordion.asp.html>

Dr. Zhang Ting’s Faculty Information:

<https://www.uhd.edu/academics/sciences/computer-science-engineering-technology/Pages/bio-zhangtin.aspx>

**Appendix**

[**Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.sjvny64gp0dy)

[**Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .3**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.eor1wtkauk6e)

[**HTML design / CSS design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.h7wyj892ctj3)

[**Javascript design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .9**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.sco2ui2wi4i7)

**Update and maintenance instructions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12**

[**Database . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.n65sn8z3mgtm)

[**Hosting and Deployment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .15**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.dr7h7nu899gq)

[**Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.gjdgxs)

[**Future Works . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.tavn7uheq2i8)

[**References . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20**](https://docs.google.com/document/d/1Kms3fhYypBkAbRPdl6TpdV5i43P1vW87WktCq-E0DL8/edit#heading=h.7escjlypprld)