Introduction

The capstone assignment is your culminating project for the Web Development program. The purpose is to create an application that showcases your capabilities, talents, insights, work ethic, and knowledge. As such it should be sophisticated, easy to use, polished, and professional. Don't forget accessible to people with disabilities and device independent or responsive. It is a lot of work. In order to help you focus on creating the best application we have limited the choice of applications. This is not to stifle your creativity but to give you a clearer goal in your work. Each of the applications comes with an extensive list of user stories that will also help you build the application.

First steps

Here is what you should do first.

1. Pick an application and make sure one of the faculty members approves your choice.
2. Choose a subject for your application. For example, if you choose Appointment Booking, is it for a barber shop or a tutor?
3. Read the user stories and make sure you understand them and consequently understand the complexity of the application you wish to build.
4. Determine if the general list of user stories is missing user stories that are important to the context of your subject. For example, a tutor appointment booking app might have user stories for a parent user as well as a child user.
5. Be careful – your content may also require user stories not listed.
6. Evaluate the user stories based on your understanding of what the MVP is your application. There are three criteria for the user stories:

    a. Must have – these are core user stories you will be building.
    b. Should have – these are a smaller set of user stories that add functionality but are not a part of the MVP.
    c. Nice to have – these are the stories that add finesse and sophistication to the application.

7. You will create a list of these user stories based on the above criteria.
8. For each of the Must have user stories you will evaluate and determine what it will take for you to create the functionality in that user story. This will become a part of the requirements document.

User stories

A user stories described what a user can do. As a developer you will read the user story and then determine what is needed in that user story in order to create it. It is a list of tasks or a to do list. You will do this for each of the Must Have and Should Have user stories. Along with this list of tasks you will give an hourly estimate of the time you think it will take you to do this task. Here are a couple of examples.

As a user I want to login to the Web application.

What are the components of a login:

Task	Time
Page text	0.5
Login form	
username	0.5
password	0.5
submit button	0.5
database functions	15
validation	1.5
forgot password	0.5
stay logged in	2
captcha	7
Total	28

The forgot password function is not a part of this user story – so you would create a link to that function. You would have another user story – *As a user I want to retrieve a forgotten password.*

As an admin user I want to edit paged content.

You should know that most user stories are a part of CRUD functionality. The login in is a Read function – reading the database record and making sure it is correct. This second user story is an Update function.

Task	Time
Access to pages	4

Display desired page	1.5
Submit button	0.5
Cancel button	0.5
Database functions	12
Total	18.5

One thing to note about this user story is that is it tied to another user story – *As an admin user I can create new paged content*. You can see that it would be the same form with the same form fields and the same validation. The appropriate reuse of code is a key skill for a Web developer.

These are two examples of the detail and format for what is required for all of the user stories you intend to use. Please note that these are not necessarily complete: messaging, layout and design, accessibility, and responsiveness have not been included but may have a significant impact.

If you are using WordPress or another CMS to build your application, you might find that many of your user stories have already been created. Even so – you should touch each one to make sure it works the way you want it to. Because a CMS has much of the coding in place you will have to ensure that your skills and capabilities shine through. A standard practice in many Web development agencies using WordPress is to use parent and child themes. If you create your own themes – that would be an accomplishment. Please – WordPress projects have to be complete in all aspects.

How many user stories should I have?

It is impossible to say for each application. Part of the discussion you will have with your supervising faculty will be about how big, complex, and difficult this project will be. We will want to ensure that you do not overwhelm yourself in creating this application. As a rule of thumb aim for twenty-five Must have user stories, five Should have user stories, and three Nice to have user stories. Also keep in mind what the sequence is for these user stories. In this way you want to carefully built the interactions across the application.

What else is in the requirements document?

One thing you should note about these applications is that they have no content. You are responsible for the content. The content will be text, images, external APIs, and whatever else you want. You will have to tell us about the content and how you intend to create or obtain it. You will also write an overall rationale for the application: this is the why, the goals, and what you hope to achieve. Navigation and layout will also be a part of the document. This may take you longer but in the end the document has to be complete. Here is a proposed table of contents for the requirements document.

1. Introduction and rationale.
2. Content evaluation.
    a. Navigation
    b. Layout, a wireframe at the start.
    c. Content descriptions and sources.
3. Functionality (user stories)
    a. Must have user stories.
    b. Should have user stories.
    c. Nice to have user stories.
    d. Hourly estimates.
4. Technical specifications.
    a. Technology stack.
    b. Hosting information
    c. Other third-party software (provide links and versions for everything used.)
5. Development specifications.
    a. Folder structure.
    b. File-naming conventions.
    c. Accessibility standards
    d. Responsiveness requirements
    e. Github links.
6. Schedule
7. Conclusion

Rubric

You will meet with your supervisor to go over this material. We will want to see it complete and comprehensive with all the components above in place. Think of the table of contents above as your backlog. Once they are all done – you are finished.

Marks	10	6	4	1
	All elements of the table of contents are present. A good number of user stories (30 or so)	Many elements of the table of contents are present. Gaps in user stories, wireframes, content.	Major gaps in the document. Missing content areas. Overall and incomplete document.	Almost no effort made to create this document.

	navigation and wireframes. Along with a clear plan for the content.	Development and other details may be missing.		

Finally

The purpose of these exercises and this document is to properly plan out the application before you begin coding. By developing the user stories in this way, you will create a series of to do lists – once you have these – this is your project. When all the user stories are working and incorporated into the application then you are done. Isn't that a good thing?