****

**EDU-Prep-iv 2020 Capstone Project**

**Timeline**

* **Duration: ​**10 days
* **Due Date:​**​Wednesday, 25th March-3rd April , 2020 5pm

**Preparation Guidelines**

These are the steps you ought to take to get ready to start building the project

Steps

1. **Choose one project out of five**
2. Create a ​Pivotal Tracker Board and populate with user stories from one of the project requirements below
3. Create a ​GitHub Repository, add a README, and clone it to your computer

​***TIP​:Find how to create a GitHub Repository*** *​*[*​****here***](https://www.google.com.ng/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwiYjuDNnNnVAhXlJ8AKHSacB9gQFggsMAI&url=https%3A%2F%2Fhelp.github.com%2Farticles%2Fcreate-a-repo%2F&usg=AFQjCNEkq3HBxdtnBJ1YKqccJVhFEv1Iuw)

* You are required to create UI templates (web pages) for your web app with HTML and CSS
* Do NOT download or use an already built website template.
* Do NOT use frameworks or libraries like Angular, Vue or React.
* Do NOT use any database, use arrays and objects or even browser storage for your data purposes

# Target Skills

**This challenge exercises the following skills.**

|  |  |  |
| --- | --- | --- |
| **Skill** | **Description** | **Helpful Link** |
| **Project Management** | Using a project management tool(pivotal tracker) to manage your progress while working on tasks. | ● To get started with Pivotal Tracker, use [Pivotal Tracker quick start](https://www.pivotaltracker.com/help/articles/quick_start/).  ● [Here's](https://docs.google.com/document/d/1ADcZ54o1s2aBtZ0dZ7fs_liFew1zTg10JqZxgRuNqm4/edit?usp=sharing) a sample template for creating Pivotal Tracker user stories. |
| **Version control And Git** | Using GIT to manage and track changes in your project. | ● Use the recommended [Git Workflow](https://www.atlassian.com/git/tutorials/comparing-workflows#gitflow-workflow),[Commit Message](https://chris.beams.io/posts/git-commit/) And [Pull Request (PR)](https://guides.github.com/activities/hello-world/#pr) standards. |
| **Front End Development** | Using HTML and CSS to create user interfaces.  Using javascript to create interactive user interfaces | [See this tutorial](https://www.youtube.com/watch?v=Wm6CUkswsNw) |
| **UI/UX** | Creating good UI design and user experience | ● See rules for good UI design [here](https://www.elegantthemes.com/blog/resources/10-rules-of-good-ui-design-to-follow-on-every-web-design-project)  ● See [this](https://code.tutsplus.com/tutorials/ui-design-for-developers-introduction--active-9921) article for More guide  ● For color palettes, see this [link](https://color.adobe.com/explore/?filter=most-popular&time=month) |

# **Evaluation Rubric**

**Use this as general guidelines to assess the quality of your work. LFAs will use this to give feedback on areas that should be improved on.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterion** | **Does not Meet Expectation** | **Meets Expectations** | **Exceeds Expectations** |
| **Project Management** | Fails to break down modules into smaller, manageable tasks. Cannot tell the difference between chores,bugs and features | Breaks down each module into smaller tasks and classifies them. Constantly updates the tool with progress or lack of it | Accurately, assigns points to the tasks.  Informs stakeholders of Project progress/blockers in a timely manner |
| **Version control And Git** | Does not utilize branching but commits to master branch directly instead. | Utilizes branching,pull-requests, and merges to the develop branch. Use of recommended commit messages. | Adheres recommended GIT workflow and uses badges. |
| **Front End Development** | Fails to develop specified HTML/CSS web pages or uses an already built out website template, or output fails to observe valid HTML document structure | Successfully develops HTML/CSS webpages while observing standards such as doctype declaration, proper document structure, no inline CSS in HTML elements, and HTML document has consistent markup | Writes modular CSS that can be reused through markup selectors such as class, id. Understands the concepts and can confidently rearrange divs on request. |
| **UI/UX** | Page is non responsive,elements are not proportional, color scheme is not complementary and uses alerts to display user feedback | Page is responsive (at least across mobile, tablet and desktops), color scheme is complementary, and uses properly designed dialog boxes to give user feedback | Page is responsive and Accessible |
| **Programming Logic** | The user interface is not interactive and the code does not work in accordance with the ideas in the problem definition. | The user interface is interactive and the code meets all the requirements listed in the problem definition. | The code handles more cases than specified in the problem definition.The code also incorporates best practices and optimisations. |

Project 1

EDU - Stack OverFlow

# **Project Overview**

EDU-StackOverflow is a platform where edu students and alumni can ask questions and provide responses.

**Required Features**

1. Users can create an account and log in.

2.Users can post questions.

3.Users can delete the questions they post.

4.Users can post answers

5.Users can view the answers to questions.

6.User can accept a response out of all the responses to his/her question as the

preferred answer.

**Optional Features (Extra Credit)**

1.Users can upvote or downvote an answer.

2.Users can comment on an answer.

3.User can fetch all questions he/she has ever asked on the platform

4.Users can search for questions on the platform

5.Users can view questions with the most responses.

**Project 1 Guidelines**

1.On Pivotal Tracker, create user stories to setup the User Interface elements:

a.User signup and signin pages.

b.A page/pages where a user can do the following:

i.View a list of recently asked questions on the platform

ii.View a question with all the answers posted for it and add an answer.

iii.Post a question.

c.A page/pages for a user’s profile which, at minimum displays:

i.The number of questions asked.

ii.The number of answers given.

iii.The list of questions asked by the user with the most answers.

iv.The list of recent questions asked by the user.

2.On Pivotal Tracker, create stories to build out your front-end with vanilla Javascript.

3.On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be a feature, bug or chore for this challenge.

4.Create a directory called UI in your local Git repo and build out all the necessary html and css pages specified above.

5.Implement interactivity with javascript on another branch

6.Deploy your interactive front-end app to gh-pages

7.Use dummy data where necessary.

**Project 2**

**EDU - Restaurant Menu**

# **Project Overview**

EDU-Restaurant menu is an online platform where users can order for different types of foods and drinks and get responses whether the chosen type of food of their choice is available or not.

**Required Features**

1. Users can create an account and log in.

2. Users can order for a certain type of food.

3. Users can see a list of all foods.

4. Users can see a list of all drinks.

5. Users can search for available foods on the menu.

6. Users can add or remove a certain type of food from the menu.

**Optional Features (Extra Credit)**

**Project 1 Guidelines**

1.On Pivotal Tracker, create user stories to setup the User Interface elements:

User signup and signin pages.

2.On Pivotal Tracker, create stories to build out your front-end with vanilla Javascript.

3.On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be a feature, bug or chore for this challenge.

4.Create a directory called UI in your local Git repo and build out all the necessary html and css pages specified above.

5.Implement interactivity with javascript on another branch

6.Deploy your interactive front-end app to gh-pages

7.Use dummy data where necessary.

**Project 3**

EDU - Video Library

# **Project Overview**

EDU-Video Library is a platform where students can pick different movies to watch.For a student to take a movie from the library they should have presented their ids and are given 3 days to return the movie back to the library …..

**Required Features**

1. Users can create an account and log in.

2. Users can add or remove movies in a cart.

3.Users can search for movie in the library

**Optional Features (Extra Credit)**

1. Users can like a movie.
2. Users can comment on movies.

**Project 1 Guidelines**

1.On Pivotal Tracker, create user stories to setup the User Interface elements:

User signup and signin pages.

2.On Pivotal Tracker, create stories to build out your front-end with vanilla Javascript.

3.On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be a feature, bug or chore for this challenge.

4.Create a directory called UI in your local Git repo and build out all the necessary html and css pages specified above.

5.Implement interactivity with javascript on another branch

6.Deploy your interactive front-end app to gh-pages

7.Use dummy data where necessary.

**Project-4**

**EDU - Book Library**

# **Project Overview**

EDU-Book Library is a platform where students who like reading pick different books and for one to pick a book from the library they must be either an EDU student, EDU LF and a EDU staff member.

**Required Features**

1. Users can create an account and log in.

2. All users can see available books in the library.

1. Books belong to particular categories ie Technology, Science,Mathematics etc.
2. EDU LF should be able to add a new book to the library.

5. Only an EDU student, EDU LF and a EDU staff member are eligible to take a book from the Library.

**Optional Features (Extra Credit)**1. Only the EDU LF should be able to delete a book.

2. EDU students, LFs and staff members should be able to search for a particular book using the book title.

3. Change status of book to **Borrowed** when book is taken by student and status to **Available** when book if not borrowed.

**Project 1 Guidelines**

1.On Pivotal Tracker, create user stories to setup the User Interface elements:

User signup and signin pages.

2.On Pivotal Tracker, create stories to build out your front-end with vanilla Javascript.

3.On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be a feature, bug or chore for this challenge.

4.Create a directory called UI in your local Git repo and build out all the necessary html and css pages specified above.

5.Implement interactivity with javascript on another branch

6.Deploy your interactive front-end app to gh-pages

7.Use dummy data where necessary.

**Project - 5**

**EDU - Online Shopping**

# **Project Overview**

EDU-Online shopping is an online platform where people can buy and sell different items and goods.

**Required Features**

1. Users can create an account and log in.

2. Users can order for a particular item.

3. Users can search for particular items.

4. Users can remove items for the cart

5. Users can add items in the cart.

**Optional Features (Extra Credit)**

**Project 1 Guidelines**

1.On Pivotal Tracker, create user stories to setup the User Interface elements:

User signup and signin pages.

2.On Pivotal Tracker, create stories to build out your front-end with vanilla Javascript.

3.On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be a feature, bug or chore for this challenge.

4.Create a directory called UI in your local Git repo and build out all the necessary html and css pages specified above.

5.Implement interactivity with javascript on another branch

6.Deploy your interactive front-end app to gh-pages

7.Use dummy data where necessary.

|  |  |
| --- | --- |
| **Project Concepts** | **Tips** |
| **HTML** | Use index.html as your entry point.  Separate your stylesheet and script file from the markup by using **external** CSS and JS files  Put all Images in a folder |
| **CSS** | Use the color theory to get complimentary and relevant color codes for your web app |
| **Javascript requirements** | Make use of the search algorithm if you have a catalogue functionality on a page.  Implement CRUD operations on the DOM  Use arrays and objects to handle data  Use classes for your algorithms |
| **Applied visual design** | Typography:Use at least 3 different fonts that complement each other in your app |
| **Mobile responsiveness** | Use media queries to make your web app run responsively on all mobile devices. |
| **Layout** | Make use of cards layout with flexbox and grid system |
| **Accessibility** | Use accessible colors for people who might be color blind and provide and use the ‘alt’ attribute on all your images |