Final Capstone

Full Stack Application

merit AMERICA

Final Capstone

Here you will learn what it is like to build software as part of a team. You may even learn something new that we didn't cover in class! You will produce a working application from the ground up that puts what you have learned into perspective.

If there is anything to keep in mind during the final capstone, this is not about one's technical prowess. This experience is about learning what it is like to work on a software development project. Throughout this experience, you will:

- Learn how work is prioritized and worked on by the team
- Participate in daily meetings to share your status
- Collaborate closely with each other and a Product Owner to develop functional software

Above all, the teams that work the best during the final capstone communicate frequently and look for opportunities to share the workload.

merit AMERICA

Agile Methodology

During the capstone, you will implement a software development methodology known as Agile. Agile is an iterative approach to managing software projects.

Agile values individuals and interactions, working software, customer collaboration, and the ability to respond to change.

Agile & Scrum

Your capstone teams will utilize a framework of agile known as Scrum. Within Scrum, there will be a series of events to adhere to. These are:

- **A Sprint**. A sprint is a limited time in which the team's focus is to develop some portion of functionality within the product scope that can be released. *Anticipate 3 day sprints.*
- **Sprint Planning**. A highly collaborative event at the beginning of a sprint in which work for the upcoming sprint is discussed and broken down into manageable tasks.
- **Daily Stand-Up/Scrum**. A 15-minute meeting at the start of the day for the team to convene and discuss work completed, upcoming work, and any roadblocks. The team identifies all roadblocks as early as possible so that the Scrum Master can help resolve them.
- **Sprint Review/Demo**. A ceremony after each sprint. The team will demonstrate any developed functionality to the Product Owner, who deems the features as "Done."
- **Sprint Retrospective**. An opportunity after the Sprint Review for the team to inspect itself and identify a plan for improvements during the next sprint.

Team

Three primary roles make up each team:

- **Product Owner**. The Product Owner is responsible for defining the scope and the prioritization of features on the project. ***This will be a combination of your Tech Squad Instructor and Technical Coaches***.
- **Development Team**. The development team (you) will consist of cross-functional professionals (e.g., BA, QA, Developer, etc.) capable of performing the work necessary to release a working version of the product. The development team collectively works together towards completing tasks within a given sprint.
- **Scrum Master**. The Scrum Master is responsible for ensuring that the team adheres to the theory behind Scrum. The Scrum Master helps the team eliminate roadblocks and minimizes the work in progress. ***One team member will be chosen as the Scrum Master (aka Team Lead).***

Within the Development Team, there is no concept of a title. All team members are responsible for understanding the requirements, developing features, and providing adequate testing.

There may be specialty skills and focus areas, but accountability ultimately belongs to the Development Team as a whole.

Schedule

Week 1	Week 2	Week 3
Sprint 1 (Design) - Choose project - Create wireframe - Agree on UI Framework (Reactstrap, Ant Design, Styled Components, etc…) - Demo to Technical Coaches for feedback/support **Sprint 2 (Baseline)** - Create Backend repository - Create Frontend repository - End-to-end functionality for authentication, storing token in Redux - Demo to Technical Coaches for feedback/support	**Sprint 3 (Functional)** - Distribute 1-3 user stories to each team member - Demo to Technical Coaches for feedback/support **Sprint 4 (Functional)** - Distribute 1-3 user stories to each team member - Demo to Technical Coaches for feedback/support	**Sprint 5 (Bug Fixing / Polishing)** - Test, Test, Test - like you're going to demo - Demo to Technical Coaches for feedback/support **Sprint 6 (Presentation)** - Develop slides for a presentation with a demo - Practice, practice, practice both individually and as a group **Presentation**

Starter Code

As with previous Capstones, you will be provided some starter code for both the React front end as well as the Spring Boot backend which includes authentication.

You will be responsible for forking these, similar to the Pair Programming assignments.

User Stories

Requirements for the project will come in the form of **user stories**.

A user story is a short, simple description of a feature told from the person who desires the new capability, usually a customer of the system.

User stories are vague so that the team can openly discuss the functionality and expectations of each feature before deciding on a “definition of done.”

User Stories

Consider the following user story example:

As a cookout host, I can add attendees to the cookout so that they will see it on their upcoming event schedule.

A potential “definition of done” may include:

- Hosts can add attendees to a new event.
- Hosts can view all attendees for an event.
- Hosts have the option to update an event by adding new attendees.
- Attendees can see events on their schedules.

While you determine your “definition of done” during Sprint Planning sessions, you may find that you need to break down a user story into smaller tasks. Breaking stories down is acceptable, and your Scrum Master will assist you with this.

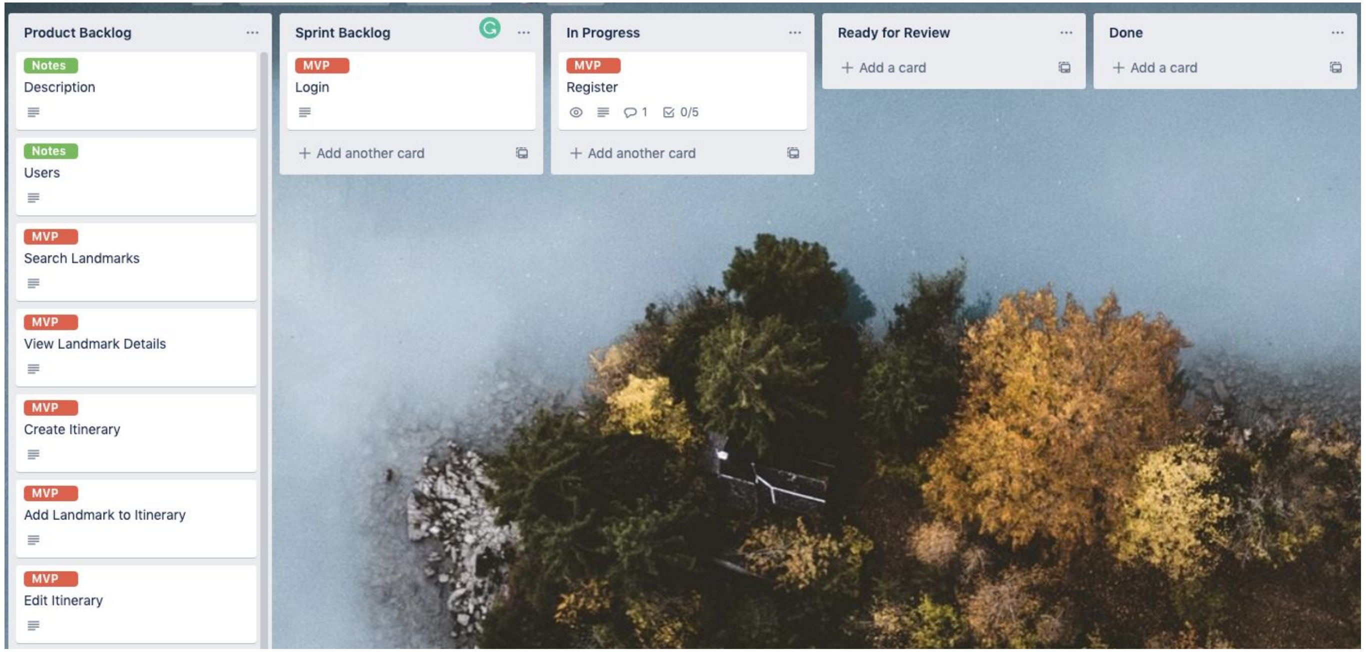
The development team should only complete work that is associated with a user story.

Managing Work with Trello

We will demo this shortly

It is the responsibility of the Development Team to manage status and work in progress (WIP). On an actual project, the Scrum Master and Product Owner might be non-technical. Therefore they may not be interested in technical updates (e.g., "I still need to create the controllers"). These team members will rely on project management tools to view the high-level status of your project. One such tool is Trello.

Each team will receive access to Trello. The Scrum Master will expect the team to update its status daily.


Product Backlog
Notes
Description
Notes
Users
MVP
Search Landmarks
MVP
View Landmark Details
MVP
Create Itinerary
MVP
Add Landmark to Itinerary
MVP
Edit Itinerary

Sprint Backlog
MVP
Login
+ Add another card

In Progress
MVP
Register
1 0/5
+ Add another card

Ready for Review
+ Add a card

Done
+ Add a card

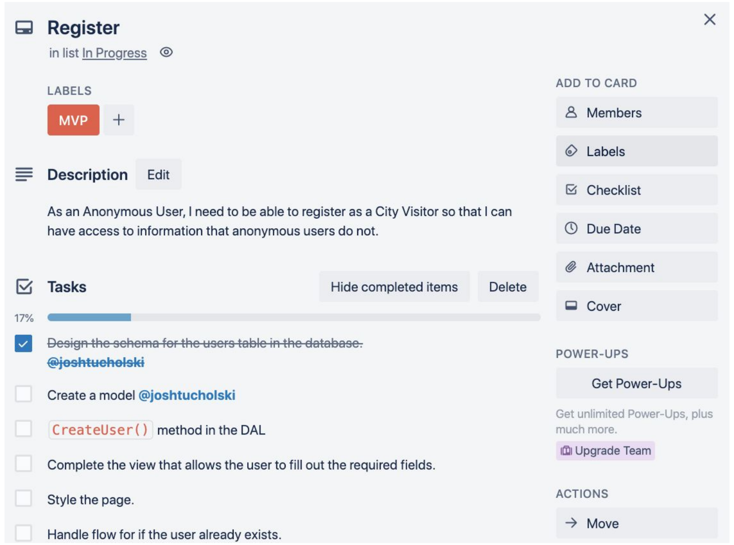
Managing Work with Trello

Your Trello board will come pre-configured with a few lists. These lists will include cards (the user stories for your project) that can be collaborated upon and moved from list to list.

- **Product Backlog**. The Product Backlog contains a prioritized list of cards representing the desired scope for the project.
- **Sprint Backlog**. The Sprint Backlog represents committed work for the current Sprint. All work in the Sprint Backlog includes a "definition of done" that the team has seen. Do not accept any work into the Sprint Backlog that lacks a definition of done.
- **In Progress**. Any feature currently being worked on should be in the In Progress list. The team should strive to limit the number of features in progress. Before taking new tasks from the Sprint Backlog, look to see if you can help a team member complete their work in progress.
- **Ready for Review.** Ready for Review signifies that development work on the feature appears complete and requires a final review from the Product Owner. In some cases, the Product Owner may determine the user story does not meet the "definition of done" and place the user story back to In Progress.
- **Done**. Done represents features that are 100% complete and have met the "definition of done." Done includes testing, styling and can be demonstrated. If there are any loose ends or "finishing touches," then it is not done. Only the Product Owner can determine if a user story is done.

Managing Work with Trello

When working on a feature, you will likely need to track tasks that pertain to a user story. With Trello, you can **create subtasks** associated with a card. Use these to follow some of the detail-level work items necessary to complete the feature.



Register

in list In Progress

LABELS

MVP

Description Edit

As an Anonymous User, I need to be able to register as a City Visitor so that I can have access to information that anonymous users do not.

Tasks

Hide completed items | Delete

17%

- [x] ~~Design the schema for the users table in the database. @joshtucholski~~
- [ ] Create a model @joshtucholski
- [ ] `CreateUser()` method in the DAL
- [ ] Complete the view that allows the user to fill out the required fields.
- [ ] Style the page.
- [ ] Handle flow for if the user already exists.

After each day, review the work completed and make sure that your Trello board reflects your team's status.

Sprint and Final Demo Tips

During each demo, the team will meet with the Product Owner to show the completed work from the sprint. With the approval of the Product Owner, they will ceremoniously move the user story into the Done column. When preparing for your Sprint Review and Final Demonstration, here are some essential things to keep in mind.

1. **Script your demo.** Create a believable walkthrough of your application.
2. **Practice your demo.** Practice giving your demo. Make sure everything works.
3. **Get everyone involved.** Software demos are something you'll likely need to do. It can be uncomfortable. Practice now.
4. **Explain features, not code.** Your Product Owner doesn't understand DAOs or CSS. They understand features. Practice explaining your project to non-technical family or friends.
5. **Use believable data.** john@meritamerica.org is much more credible than abcdadsfasdf@xyz.com.
6. Clear your autofill history.

Demo Tips

7. **Don’t talk and type.** Have one team member explain the feature while another demonstrates how it works.
8. **Capture feedback.** If your Product Owner mentions, “it would be nice if…”, capture it and associate it with a card. Then ask if it is required or “nice to have” to help determine how to prioritize the work.
9. **Don’t pull master right before the demo.** Have a team member who has a stable build on their machine be prepared an hour or two before the demo.
10. **Ask early for feedback.** If you want to know right away if you are on the right path, reach out to your Product Owner and offer a short demo or a screenshot.

Never make “one last change.” Sooner or later, it backfires.

Communication

Communication between all team members is essential. Each team will have access to these tools to aid in communication:

- **Chat**. Use Slack group chat to communicate within your teams. You can also reach out to your instructors for guidance in between live sessions.
- **Video Conferencing.** Team Lead will send invites for all Stand-Ups, Sprint Reviews, and Planning sessions as well as working sessions.
- **Pair Programming/Screen Sharing.** We suggest using Zoom for screen sharing. With a Basic (free) account, you can have unlimited-length meetings between two participants. If you have two or more participants, your meeting will be limited to 40 minutes.
- **Whiteboarding.** Sketchboard offers a decent collaborative application for whiteboarding together.
- **Project Status.** As indicated, use Trello to track the team's work in progress.
- **Live Sessions.** Come prepared to your Tech Squads and Technical Small Group Sessions prepared with questions about project requirements and technical issues to resolve as well as ready to demo your sprint/feature for feedback.

Communication

Agree upon daily Working Hours for your team (e.g. 9 AM to 5 PM). Hold each other accountable to show up. If this doesn't work for you due to personal commitments, address it with your Team Lead and develop a solution for you to make up the work.

Be responsive. You should be online during the working hours for your team to be available to support the sprint. In general, you should be monitoring Slack and showing up to meetings to respond to messages to keep making progress.

Escalate problems early. Fifteen days is a small window of time to develop a working product. If you know that there will be an issue, the sooner it is identified as a risk, the sooner the Scrum Master can help mitigate that risk.

Above all, communicate. Agile favors individuals and interactions over processes and tools.

Professionalism

Treat this capstone project as your first day on the job. Your peers are your team, and the Product Owner is your customer.

OUR EXPECTATIONS

We expect you to participate in meetings and contribute to the delivery of your project.

We expect you to be accessible to your team and Merit America during working hours.

We expect you to communicate changes that may affect your ability to deliver the project.

We expect you to look and act professionally while on camera during all Scrum meetings.

We expect you to work for the good of the group by leaving your ego at the door.

We expect you to remain open to new ideas and learn what it is like to work on a software development project.

Final Advice

Finally, here are some general pieces of advice to keep in mind throughout the project:

- **Make a plan.** Identify the main screens, the database schema, the user flow and review it with your Product Owner. Make sure it is clear how the work you are doing fits into the plan.
- **Agree upon your architecture.** As a team, determine how you want to build your application. Do you want to use an MVC framework (i.e., Spring MVC) or RESTful architecture with a JavaScript front-end? Consider if you will need to host your application somewhere.
- **Commit early; commit often**. Commit at least before lunch and the end of the day.
- **Pull early; pull often**. Avoid working with out-of-date code by checking for updates.
- **Mob early on**. Bring the whole team together early on to develop familiarity with the codebase at the same time.
- **Communicate frequently to avoid merge conflicts**. Avoid duplicating work or erasing your teammate's work by communicating what you are working on.

Final Advice

- **Review each other's code**. Become familiar with other parts of the system. You may need to jump in and troubleshoot or help your team fix bugs later.
- **Avoid spinning your wheels.** The final capstone goes by fast. Ask someone on your team to pair with you when you get stuck. Reach out to an instructor to talk through problems.
- **Use the frameworks and languages we teach at Merit America**. You may use libraries and APIs that we didn't cover, but please do not take on a new framework (i.e., Angular, Swift). Two weeks does not provide enough time for the team to learn a new framework or language. Furthermore, the academic staff likely won't support you when you are stuck.
- **Limit your work in progress (WIP)**. The entire team should be working towards completing in-progress user stories. Fewer completed features make for a better product than a lot of partially completed features.
- **Script your database**. Script all of the DROP and CREATE statements for your database. You will continuously rebuild your database throughout the project.

You've worked hard to get to this point. Building software is the fun part.

Project Choices

merit
AMERICA

General Project Details

All projects have:

- Required features (user stories)
- Optional features (user stories)
- Authentication (register new users and login existing users)
- Full Stack: Front-end (React) and Back-end (Spring+PostgreSQL)

User Stories may have varying levels of detail. As a team, you have the flexibility to interpret the requirements as you envision your application. You also have the opportunity to reach out to your technical coaches for guidance (or to break a tie) in the event your team cannot decide on an interpretation.

Restaurant Tinder

This application helps people make a decision which restaurant they should visit based upon their preference.

Features:

- Register/Login
- Invite friends out to eat at specific restaurants
- Guests can vote with thumbs up/down
- View restaurant finalists

Meal Planner

Meal planning is hard. You have to come up with what to make based on what is in your pantry, make it to the store, and then keep track of all of the different ways to prepare the food. The meal planner application allows users to create their own library of recipes and plan meals for the week using existing recipes (and ingredients).

Features:

- Create recipes
- Add ingredients to recipes
- Create a meal plan made up of recipes
- Generate a grocery list

City Tours

Visitors will use an application to map out a tour of the city based on the best route from their hotel/starting point. If they wish, they can plan multiple routes on different days from the same location. The app will offer the most efficient route to see as many local landmarks as possible; ideal for the visitor who is just "stopping in" and does not have a feel for the lay of the land but wants to see some of the city in a short amount of time.

Features:

- Create itinerary
- Add landmarks to itinerary
- Generate travel route
- Rate landmarks (thumbs up/down)

Brewery Finder

This application shows information about breweries and beers. Beer lovers can view information about breweries and beers, while brewers can update the information about their products. Administrators can add breweries to the system.

Features:

- Brewers
  - Create Brewery
  - Add Beer to Brewery
  - Manage Brewery News and Events
- Consumers
  - View Brewery’s Beer List
  - View Beer information
  - Review Beer
  - Like a Brewery / List of Favorite Breweries

Trello Demo

merit
AMERICA