

Hands-on Lab: Complete User Stories

Estimated time needed: 60 minutes

Project overview

As a full-stack developer in the GiftLink project, your role is to develop both the backend and frontend components of the application. While the initial structure of GiftLink's backend services exists, they are not yet complete. Your task involves further developing these backend services, ensuring they are robust, secure, and scalable. Alongside backend development, you will also create the front-end application. This front end will serve as the primary user interface for the GiftLink community, designed to provide a seamless and engaging user experience. The first step in this journey is to convert the requirements into user stories the team can work on together.

User stories

In this lab, you will build a sprint plan for the project. User stories are a fundamental component in Agile methodologies, serving as a simple yet powerful tool for understanding end-user requirements. They are typically written from a user's perspective and focus on what they need from the system to achieve a specific goal. A well-crafted user story is concise and focuses on a single functionality or feature, helping to break down complex projects into manageable work units. In this lab, you'll have the opportunity to apply these principles to create and refine user stories.

Note: For this lab, you will not be using the lab environment, but will work directly in GitHub.

Objectives

In this lab, you will:

- Establish a new GitHub repository using a provided template
- Formulate a template for user stories
- Add user stories as GitHub issues
- Organize user stories in preparation for enhancing the backlog

Exercise 1: Create a GitHub repository

You will need your repository to complete the final project. We have provided a GitHub Template to create your repository in your own GitHub account. **Do not Fork the repository as it's already a template.** This action will avoid confusion when making Pull Requests in the future.

Tasks

1. In a browser, visit this GitHub repository:
<https://github.com/ibm-developer-skills-network/tqej-fullstack-capstone-template>
2. From the GitHub **Code** tab, use the green **Use this template** to create your repository from this template.
3. Select **Create a new repository** from the dropdown menu. On the next screen, fill out these prompts following the screenshot below:

The screenshot shows a GitHub repository page for 'tqej-fullstack-capstone-template'. The repository is a public template generated from [ibm-developer-skills-network/coding-project-template](#). The main interface includes a code editor with a 'main' branch, a file browser, and a commit history showing initial commits from 'ibm-skills-network-bot'. A context menu is open over the commit history, with the 'Create a new repository' option highlighted by a red box. To the right, there's an 'About' section for the template repository, which has 0 stars, 1 watcher, and 0 forks. Below that is a 'Releases' section indicating no releases have been published.

tqej-fullstack-capstone-template Public template

generated from [ibm-developer-skills-network/coding-project-template](#)

main Go to file Add file ▾ Code ▾ Use this template ▾

Create a new repository

Open in a codespace

ibm-skills-network-bot Initial commit cb0c4a2 4 m

.gitignore Initial commit 4 minutes ago

LICENSE Initial commit 4 minutes ago

README.md Initial commit 4 minutes ago

README.md

coding-project-template

About

fullstack-capstone-template

Readme

Apache-2.0 license

Activity

0 stars

1 watching

0 forks

Report repository

Releases

No releases published

[Create a new release](#)

1. Select your GitHub account from the drop-down list.
2. Name the new repository: `fullstack-capstone-project`
3. (Optional) Add a description to let people know the purpose of the repo.
4. Make the repo **Public** so that others can see it (and grade it).
5. Select **Create repository** to create the repository in your GitHub account.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Repository template



ibm-developer-skills-network/tqqej-fullstack-capstone-template ▾

Start your repository with a template repository's contents.

Include all branches

Copy all branches from ibm-developer-skills-network/tqqej-fullstack-capstone-template and not just the default branch.

Owner *



captainfedoraskillup ▾

Repository name *

/ fullstack-capstone-temp

✓ fullstack-capstone-template is available.

Great repository names are short and memorable. Need inspiration? How about [fuzzy-telegram](#) ?

Description (optional)

Repository for capstone project

 **Public**

Anyone on the internet can see this repository. You choose who can commit.

 **Private**

You choose who can see and commit to this repository.

(i) You are creating a public repository in your personal account.

Create repository

You may use GitHub's built-in kanban board for managing your project.

Alternatively, if you have a zenhub account and prefer to use Zenhub Kanban board, you can connect your GitHub repository to your ZenHub Account and optionally install the ZenHub plugin in your browser.

Exercise 2: Create a user story template

In this exercise, you will create a user story template in GitHub to help you write well-formatted user stories for your kanban board.

Steps to complete

1. Create an issue template in GitHub for your project's GitHub repository. Ensure the template includes the components listed below. You may want to copy, paste, and edit this text because it contains the correct markdown syntax you will need for the template. You can learn more about creating the issues template using the GitHub UI in the **hints** section below.

```
**As a** [role]
**I need** [function]
**So that** [benefit]
### Details and Assumptions
* [document what you know]
### Acceptance Criteria
gherkin
Given [some context]
When [certain action is taken]
Then [the outcome of action is observed]
```

Hints

▼ Click here for a hint.

1. To set up the template for user stories, click settings in your repository and then Set up templates.

Issues

Issues integrate lightweight task tracking into your repository. Keep projects on track with issue labels and milestones, and reference them in commit messages.

Get organized with issue templates

Give contributors issue templates that help you cut through the noise and help them push your project forward.

[Set up templates](#)

2. Add a custom template



captainfedoraskillup / fullstack-capstone-project



<> Code

● Issues

↑ Pull requests

▶ Actions

Projects

Wiki

! Security

...

Propose changes

Add template: select ▾

Bug report

Standard bug report template

Feature request

Standard feature request template

Custom template

Blank template for other issue types

3. Preview and edit the template

The screenshot shows a GitHub repository page for 'captainfedoraskillup / fullstack-capstone-project'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and a three-dot menu. A green 'Propose changes' button is visible on the right. Below the navigation, there's a section titled 'Custom issue template' with a placeholder text 'Describe this issue template's purpose here.' To the right of this text are two buttons: 'Preview and edit' (highlighted with a red box) and a trash can icon. At the bottom, there's a dashed-line box containing the text 'Add template: select ▾'.

Custom issue template
Describe this issue template's purpose here.

Preview and edit

Add template: select ▾

4. Click on the pencil button to edit the template



captainfedoraskillup / fullstack-capstone-project



<> Code

● Issues

⬆️ Pull requests

▶ Actions

Projects

Wiki

🛡 Security

...

Propose changes

Custom issue template

Describe this issue template's purpose here.

[Close preview](#)



Issue: Custom issue template

Describe this issue template's purpose here.

No template content.

Optional additional items

Issue default title:

Assignees:

Labels:

5. Fill in the form with the actual template provided to you.

User Story

This template defines a user story

[Close preview](#)



Template name

User Story

This file lives in **custom.md**

About

This template defines a user story

Template content

Styling with Markdown is supported

```
**As a** [role]  
**I need** [function]  
**So that** [benefit]
```

```
### Details and Assumptions  
* [document what you know]
```

```
### Acceptance Criteria  
gherkin  
Given [some context]  
When [certain action is taken]  
Then [the outcome of action is observed]
```

Optional additional items

Issue default title

This will be suggested as the issue title

Add a placeholder for issue title, ex. [BUG]

6. Propose and commit changes.

Screenshot of a GitHub repository page for "captainfedoraskillup / fullstack-capstone-project". The page shows a "User Story" template with a "Propose changes" button highlighted by a red box.

The GitHub interface includes the following elements:

- Header: Repository name "captainfedoraskillup / fullstack-capstone-project", search bar, and various navigation icons.
- Nav Bar: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and a three-dot menu.
- Main Content Area:
 - A "User Story" template card with the title "User Story" and the subtext "This template defines a user story". It includes "Preview and edit" and delete buttons.
 - A large dashed box area containing the text "Add template: select ▾".
- Footer: A note "7. Commit changes as follows:".

7. Commit changes as follows:

The screenshot shows the GitHub repository settings page for 'captainfedoraskill... / fullstack-capstone-project'. The 'Settings' tab is selected. On the left, there's a 'User Story' template section with a preview button and a trash icon. Below it is a dropdown for 'Add template: select ▾'. On the right, the 'Commit changes' section is displayed. It includes a 'Commit message' field with a placeholder 'Update issue templates', an 'Extended commit message' field containing 'Added user stories template for issues', and two radio button options for committing: 'Commit directly to the main branch.' (selected) and 'Create a new branch for this commit and start a pull request.' At the bottom is a large green 'Commit changes' button.

User Story

This template defines a user story

Preview and edit

Add template: select ▾

Commit changes

Commit message

Update issue templates

Extended commit message

Added user stories template for issues

-o- Commit directly to the `main` branch.

↗ Create a **new branch** for this commit and start a pull request.

Commit changes

8. This should create a new directory in the repository's main branch.

 captainfedoraskill... / fullstack-capstone-project | >_ | + | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Updated issue templates for this repository X

 **fullstack-capstone-project** Public Pin Unwatch 1 Fork 0 Star 0

generated from [ibm-developer-skills-network/tqej-fullstack-capstone-template](#)

main 1 branch 0 tags [Go to file](#) [Add file](#) [Code](#)

 captainfedoraskillup Update issue templates ... e203366 now 2 commits

 .github/ISSUE_TEMPLATES Update issue templates now

 .gitignore Initial commit 23 minutes ago

 LICENSE Initial commit 23 minutes ago

 README.md Initial commit 23 minutes ago

README.md Edit

coding-project-template

About 

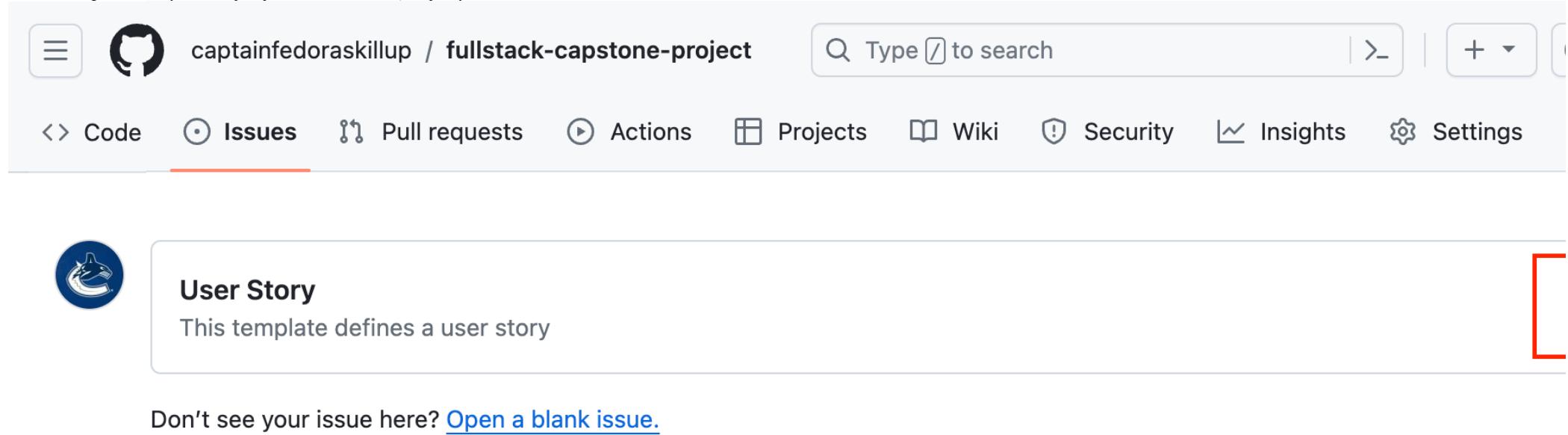
No description, website, or topics provided.

 Readme
 Apache-2.0 license
 Activity
 0 stars
 1 watching
 0 forks

Releases

No releases published [Create a new release](#)

9. When creating a new issue, you will be prompted to use the User Story template you created.



The screenshot shows a GitHub repository page for 'captainfedoraskillup / fullstack-capstone-project'. The 'Issues' tab is selected, indicated by a red underline. A card titled 'User Story' is displayed, containing the text 'This template defines a user story'. Below the card, a link says 'Don't see your issue here? [Open a blank issue.](#)'.



Exercise 3: Create new labels and user stories

You need three labels for the rest of the lab:

- *new* - The stories you need to prioritize.
- *backlog* - The stories picked up in the current sprint.
- *icebox* - The stories that don't need to be worked on immediately, but you will work on later.

Steps to complete

1. Open the labels page for your GitHub repository. You can use this URL after replacing the REPO_URL with your repository URL https://REPO_URL/issues/labels. You will use the New Label button to create the three labels.

A screenshot of a GitHub repository interface for the project "captainfedoraskillup / fullstack-capstone-project". The "Issues" tab is selected. At the top right, there is a red box highlighting the green "New label" button.

The page displays a list of 9 labels:

- bug**: Something isn't working
- documentation**: Improvements or additions to documentation
- duplicate**: This issue or pull request already exists
- enhancement**: New feature or request
- good first issue**: Good for newcomers
- help wanted**: Extra attention is needed

Each label entry includes a three-dot menu icon on the right.

2. Add the label new as shown below. You can use different colors.

The screenshot shows a GitHub repository page for 'captainfedoraskillup / fullstack-capstone-project'. The 'Issues' tab is selected. In the top right, there's a 'New label' button. A modal window is open, titled 'new'. It contains fields for 'Label name' (containing 'new'), 'Description' (with placeholder 'Description (optional)'), 'Color' (set to '#fbca0'), and buttons for 'Cancel' and 'Create label'. The 'Label name' field and the 'Create label' button are both highlighted with red boxes.

3. Add backlog and icebox labels similarly. The resulting page should look as follows:

The screenshot shows a GitHub repository interface for 'captainfedoraskillup / fullstack-capstone-project'. The 'Issues' tab is active. At the top, there are navigation links: Code, Issues (selected), Pull requests, Actions, Projects, Wiki, Security, and three more options represented by ellipses. Below the navigation bar are buttons for Labels, Milestones, and a search bar labeled 'Search all labels'. A green button labeled 'New label' is also present. The main area displays a list of 12 labels. Three labels ('icebox', 'backlog', and 'new') are highlighted with a red border. To the right of the labels is a 'Sort' dropdown menu. Each label entry includes a 'More options' button (three dots) and a 'Delete' button.

| Label | Description |
|---------|-------------------------|
| icebox | |
| backlog | |
| new | |
| bug | Something isn't working |

5. Now that you have the labels, start creating the stories. Simply create a new issue, pick the template you created earlier, and populate the details of the user story. Finally, add the `new` label to the story.

Exercise 4: Assemble your product backlog

In this exercise, you will create user stories based on the GiftLink Capstone Project. You need to develop several back-end services and a full front end in React.

Steps to complete

Create ten user stories in your GitHub repository, one for each of the following steps of your project:

1. Finish user stories
2. Initialize and populate MongoDB
3. Run skeleton application
4. Implement a landing page and navigation
5. Add authentication components and logic
6. Implement Gifts details page
7. Implement a search component
8. Design and implement the comments feature
9. Containerize the services and applications
10. Deploy backend and frontend

To create a user story, create a new issue and pick `User Story` as the template for the issue. Once you create the issue, fill in the details as the template requires.

Add the label `new` to all the user stories.

Exercise 5: Triage new issues

In this exercise, you will begin to conduct **Backlog Refinement** by inspecting the issues in the `New Issues` pipeline and moving them to the `Product Backlog` or `Icebox`, depending on when you plan to work on them. Moving the issues to a pipeline simply means applying the `new`, `backlog`, and `icebox` labels. Use your judgment to prioritize user stories based on when they need to be completed. For example, containerizing with Docker and deploying to Kubernetes is something you will do a few sprints from now, so it is not immediately important. An issue will only have one of the three labels at a time.

Steps to complete

1. Apply the `new` label to all issues initially if you did not apply the label when creating the stories.
2. Determine which user stories you will work on immediately and apply the `backlog` label. Remove the `new` label from these issues.
3. Move the remaining stories from `New Issues` into the `Icebox` as you will work on them later by removing the `new` label and applying the `Icebox` label.

▼ Click here for a hint.

TBD: Five stories will be implemented now, and two will be implemented later.

Exercise 6: Refine your product backlog

In this exercise, you will follow the steps of conducting a backlog refinement meeting. You will be the product owner, preparing the product backlog for your next sprint planning meeting. The goal of this preparation is to make all your stories sprint-ready.

Steps to complete

1. Make sure that all the stories in the `Product Backlog` have sufficient details to be considered "sprint ready." Pay special attention to the **Acceptance Criteria** to ensure you have defined the definition of "done."
2. In GitHub, create a label called `technical debt` with a yellow color code and add it to your repository.
3. Create a story called `research authentication in React and Express` and add the label of `technical debt`.
4. Assign labels to your stories. Remember that anything that brings value to the customer is an `enhancement`, and `technical debt` can be things developers need but provide no visible customer value.

▼ Click here for a hint.

At least one story should be labeled "technical debt."

Evidence

Make a note of the URL to your GitHub repository. You will need to submit it for the final project.

Conclusion

Congratulations! You have created a sprint plan for this capstone project. You are now ready to start implementing your full-stack application.



Skills Network