

Project 1

Due Oct 3 by 11:59pm **Points** 100 **Available** after Sep 20 at 12:01am

Overview

In this project, you will be creating an event hosting website. Users will be able to sign up for your site, login and logout. Users will also be able to host events and register to attend events.

Required Packages

- Python 3.x
- Flask
- Jinja2 Templating engine (included with Flask)
- Flask-SQLAlchemy

GitHub Classroom Link

Please click on the following link to get your project repository. <https://classroom.github.com/a/Q5-ktdDB> [\(https://classroom.github.com/a/Q5-ktdDB\)](https://classroom.github.com/a/Q5-ktdDB)

Specification

Below are the specifications for the application. I have split the specifications up based on MVC. Please read through the entire specification and submission guidelines before you start development.

Model

- User, each User must have:
 - username
 - each User must have a unique username
 - blank usernames are not allowed
 - password
 - passwords must be hashed before being saved to the database
 - blank passwords are not allowed
- Event, each Event must have:
 - title
 - blank titles are not allowed
 - description
 - start date & time

- blank starts are not allowed
- end date & time
 - blank ends are not allowed
- Relationships between User and Event
 - User can host many Events
 - User can attend many Events and each Event will have many Users attending
- Note: You are not limited to the columns listed above. Add additional columns and association table as needed to facilitate the relationships between User and Event.

Controller

- You will need a homepage that displays:
 - a list of upcoming events hosted by all users
 - if a user is logged in, display a separate list of upcoming events hosted by the logged in user
 - for both lists above, list the events in order of their start date & time
- The User will need routes/functions defined to:
 - register for the website
 - login to the website
 - logout of the website
- The Event will need routes/functions defined to allow the user to:
 - create an event with the logged in user as host
 - cancel an event
 - a user cannot cancel an event they are not hosting
 - when a user tries to cancel an event, they must be asked to confirm that they want to cancel the event before it is removed from the database
 - register to attend an event
 - a user should not be allowed to sign up to attend an event they are hosting
 - you do not need to allow users to unregister for an event in this project

Views (Templates)

- A base template must be created and extended by all other templates
- Templates need to be created for the following routes:
 - the homepage
 - user registration
 - user login
 - event creation
 - event cancellation (to act as the confirmation since JavaScript is not part of this project)
- The main menu for the site must show different links depending on if the user is logged in.
- Events on the homepage must list:
 - title
 - host

- description
- start date & time
- end date & time
- a link to allow a user to register to attend an event if the user is not the host of the event
- a cancel event link if the user is the host of the event
- Design and CSS is left up to you. The only design requirement is that the page is presented in a clear and readable manner.

Misc & Hints

- Look at db.DateTime for Event start date & time and end date & time
- You are not required to use a datetime picker for the start date & time and the end date & time form fields. A textbox with date & time format instructions will suffice
- Treat the one to many and many to many relationships between User and Event as two separate relationships
- You can use any of the examples from class for inspiration and you may use snippets from them in your code. However, ****DO NOT**** copy them in their entirety and edit them for this project

Submission Guidelines

- Place any comments you have for the grader to help them grade your assignment in the FOR_GRADER file
- Name the main (controller) file events.py
- Name the sqlite database events.db
- Do not submit any IDE files, feel free to add them to the .gitignore file under "# Editor files"
- You must be able to run your application by setting the FLASK_APP environment variable to events.py and running "flask run"
- You must be able to initialize your database by setting the FLASK_APP environment variable to events.py and running "flask initdb"
- Be sure to remember to push the latest copy of your code back to your GitHub repository before the the assignment is due. GitHub Classroom will not stop you from pushing additional commits after the deadline but it will make the latest commit before the deadline the submission that the grader will grade. It will also notify us if previous work is edited after the deadline.

Project 1

| Criteria | Ratings | Pts |
|---|---------|----------|
| User model created as described | | 5.0 pts |
| Event model created as described The Event model contains a column for title, description, start date & time and end date & time. It also does not allow blank titles, blank start date & times or blank end date and times. | | 5.0 pts |
| "User can host many Events" relationship coded and functional | | 5.0 pts |
| "User can attend many Events and each Event will have many Users attending" relationship coded and functioning | | 5.0 pts |
| User can register for website In addition to creating and saving the user. Blank usernames and passwords are rejected, usernames are unique and passwords are hashed before being saved in the database. | | 10.0 pts |
| User can login and logout of website | | 5.0 pts |
| Events can be created with the logged in user as host In addition to creating and saving the event with the logged in user as host. Blank titles, start date & times and end date & times are are rejected. | | 10.0 pts |
| Events can be canceled by the host Additionally, events cannot be canceled by users other than the host and the host is asked to confirm that they want to cancel the event. | | 10.0 pts |
| Events can be attended by users other than the host Additionally, the event host cannot attend an event they are hosting. | | 10.0 pts |
| Homepage shows list of all upcoming events | | 5.0 pts |
| Homepage displays a list of the events hosted by the logged in user if a user is logged in | | 5.0 pts |
| Events on the homepage are displayed as described | | 5.0 pts |
| A base template was created and extended by all other templates The other templates are the homepage, user registration, user login, event creation and event cancellation. | | 10.0 pts |
| The main menu options change depending on if a user is logged in | | 5.0 pts |
| Website is presented in a clear and readable manner | | 2.0 pts |
| Submission The files were named properly, there were no IDE files submitted, events.db was not submitted and "flask run" and "flask initdb" work as described. | | 3.0 pts |

| Criteria | Ratings | Pts |
|---------------------|---------|-----|
| Total Points: 100.0 | | |