Optional | Deadline: of Week 4 | Difficulty Level: Intermediate | Est. Time: 2 - 4 hrs

Assignment: Ninja Gold

Create a simple game to test your understanding of Flask, and implement the functionality below.

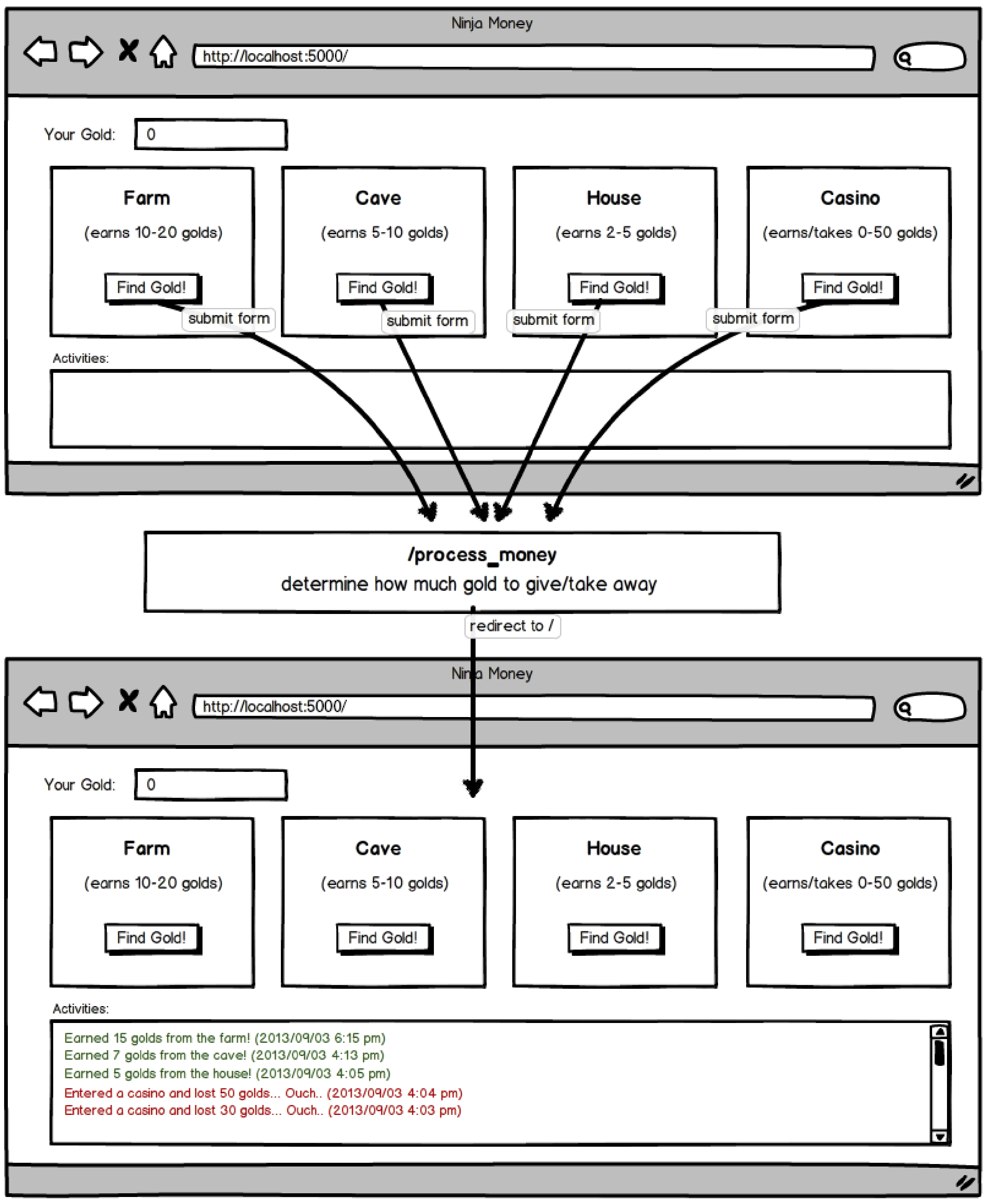
For this assignment, you're going to create a mini game that helps a ninja make some money! When you start the game, your ninja should have 0 gold. The ninja can go to different places (farm, cave, house, casino) and earn different amounts of gold. In the case of a casino, your ninja can earn *or lose* up to 50 gold. Your job is to create a web app that allows this ninja to earn gold and to display their past activities.

The root route should display the wireframe below. There should be 4 forms on the HTML page. As an example, the farm form might look something like this:

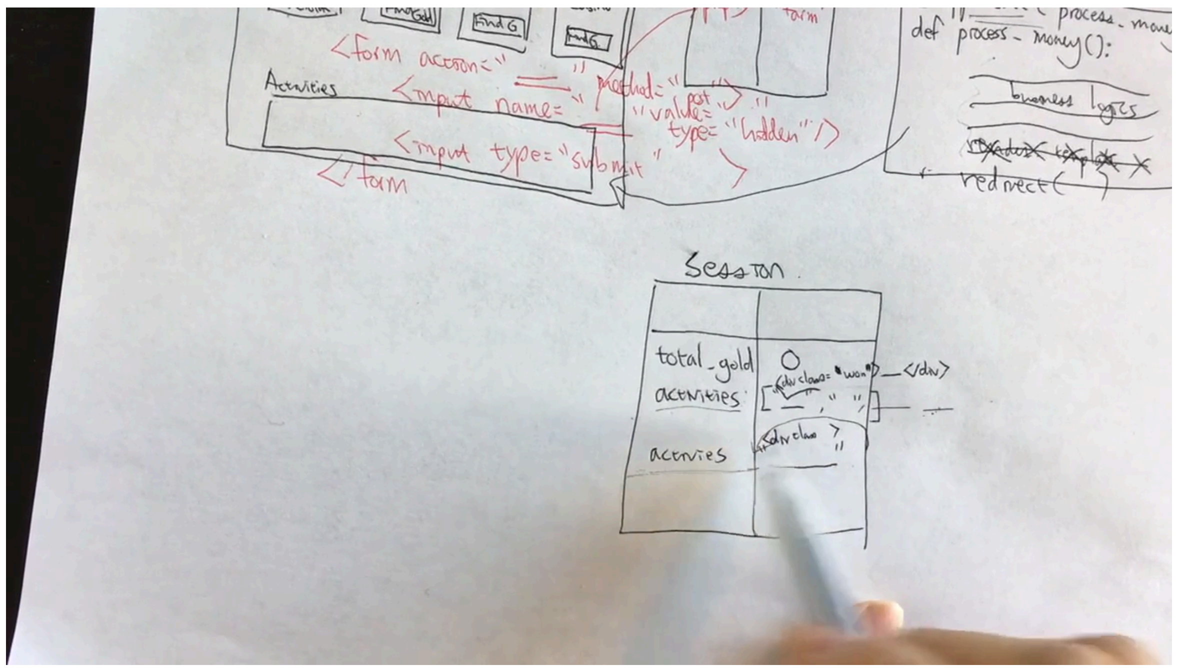
```
<form action="/process_money" method="post">
  <input type="hidden" name="building" value="farm" />
  <input type="submit" value="Find Gold!"/>
</form>
```

There should be a method that handles the POST request, determining how much gold the user should now have depending on their visit.

Note: You should only have **2 routes** for this assignment -- '/' and '/process_money'



Watch this before you start the assignment



A Helpful Tip

Consider the following code:

my_proj/server.py

```python
def index():
    message = "<ul><li>Hello</li></ul>"
    return render_template("index.html", message=message)
```

my_proj/templates/index.html

```
{{ message
}}
```

In Browser:

<ul><li>Hello</li></ul>

```
{{ message|safe }}
```

In Browser:

- Hello

By default, Jinja will convert any html entities with character entities. To prevent this from happening, we used the `safe` pipe, which you can read about in the Flask documentation and on StackOverflow.

- [ ] Create a new Flask project called ninja_gold
- [ ] Create the template as shown in the wireframe above, with 4 separate forms
- [ ] Have the root route render this page
- [ ] Have the "/process_money" POST route increase/decrease the user's gold by an appropriate amount and redirect to the root route
- [ ] NINJA BONUS: Display all the activities performed by the user in a log on the HTML, as shown in the wireframe
- [ ] NINJA BONUS: Have the activities be color-coded as shown above (+ money is green, - money is red)
- [ ] NINJA BONUS: Add a reset button to restart the game
- [ ] SENSEI BONUS: Have the activities display in descending order, with the most recent activity first
- [ ] SENSEI BONUS: Provide winning parameters to the game--for example, a user must obtain 500 gold in less than 15 moves. Only display the reset button once the user has won or lost.
- [ ] SENSEI BONUS: Complete the "/process_money" route without 4 conditional statements (i.e. without doing if farm...elif cave...etc.)

Submit

Note: Please Zip your file(s) before uploading.

Website URL File Upload

Drag & drop your files

or Browse

Or

Type in a URL here Save

Submit Assignment

Previous

Next

Privacy Policy