This is CS50

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

Today

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

Flask

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

Project Proposal

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

Test 🎉

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

Project

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

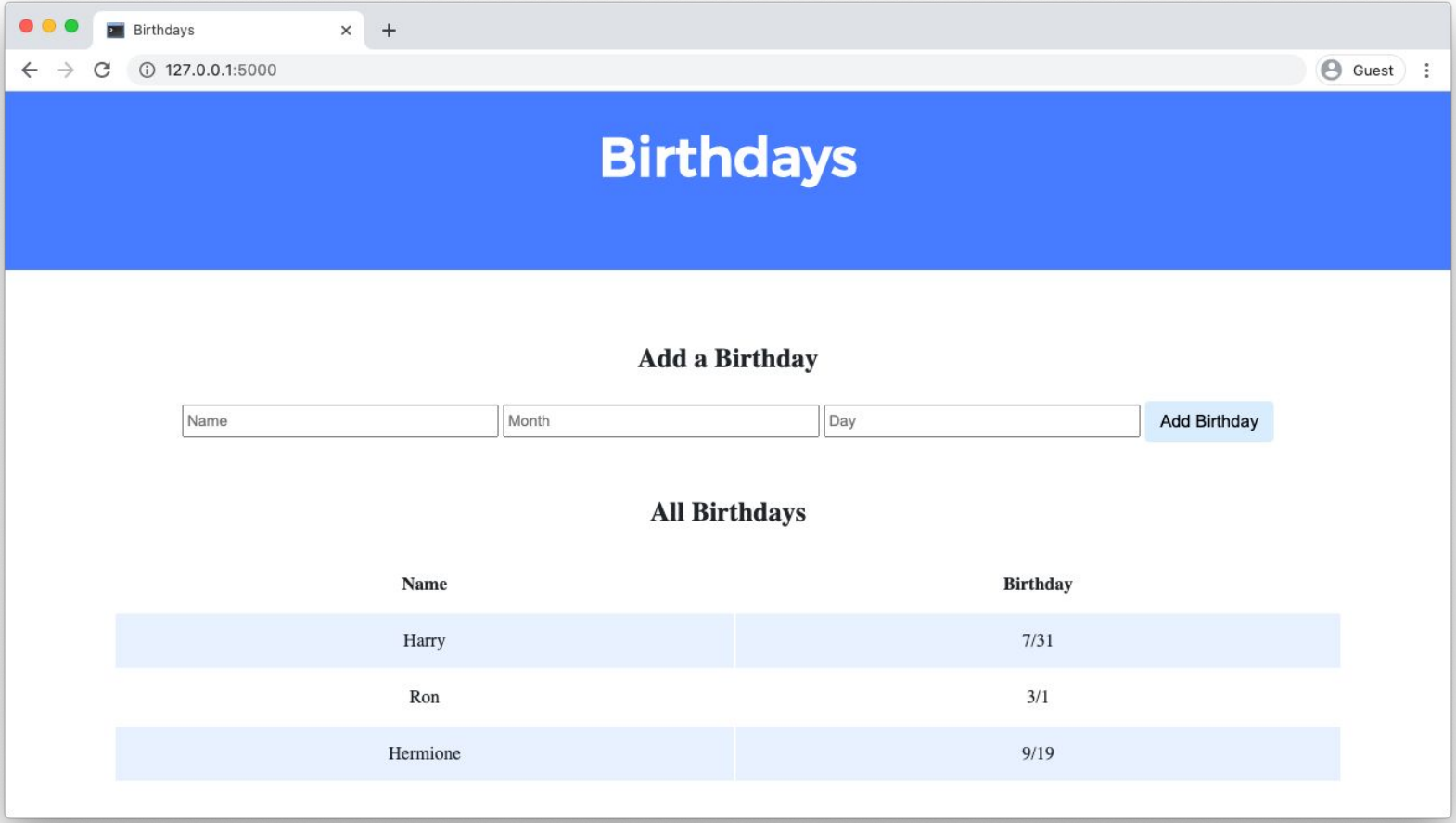
Status Report

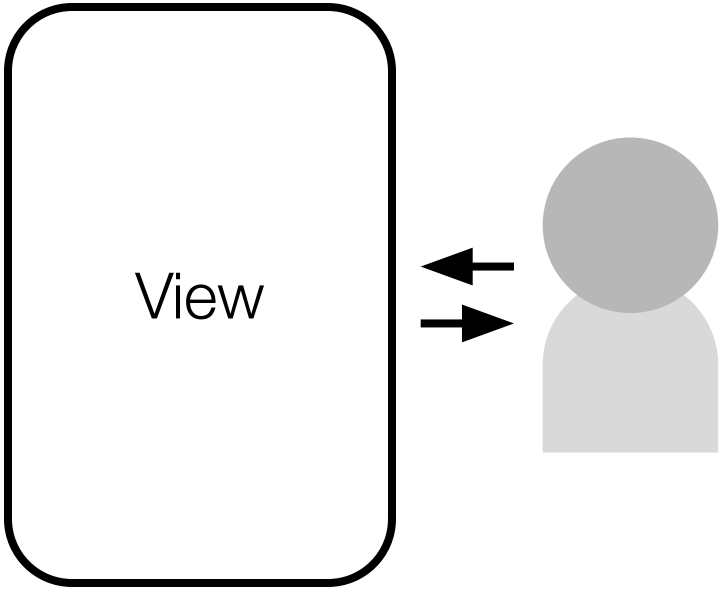
	Sat	Sun	Mon	Tue	Wed	Thu	Fri
Mar.	26	27	28	29	30	31	1
Apr.	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6
May	7	8	9	10	11	12	13

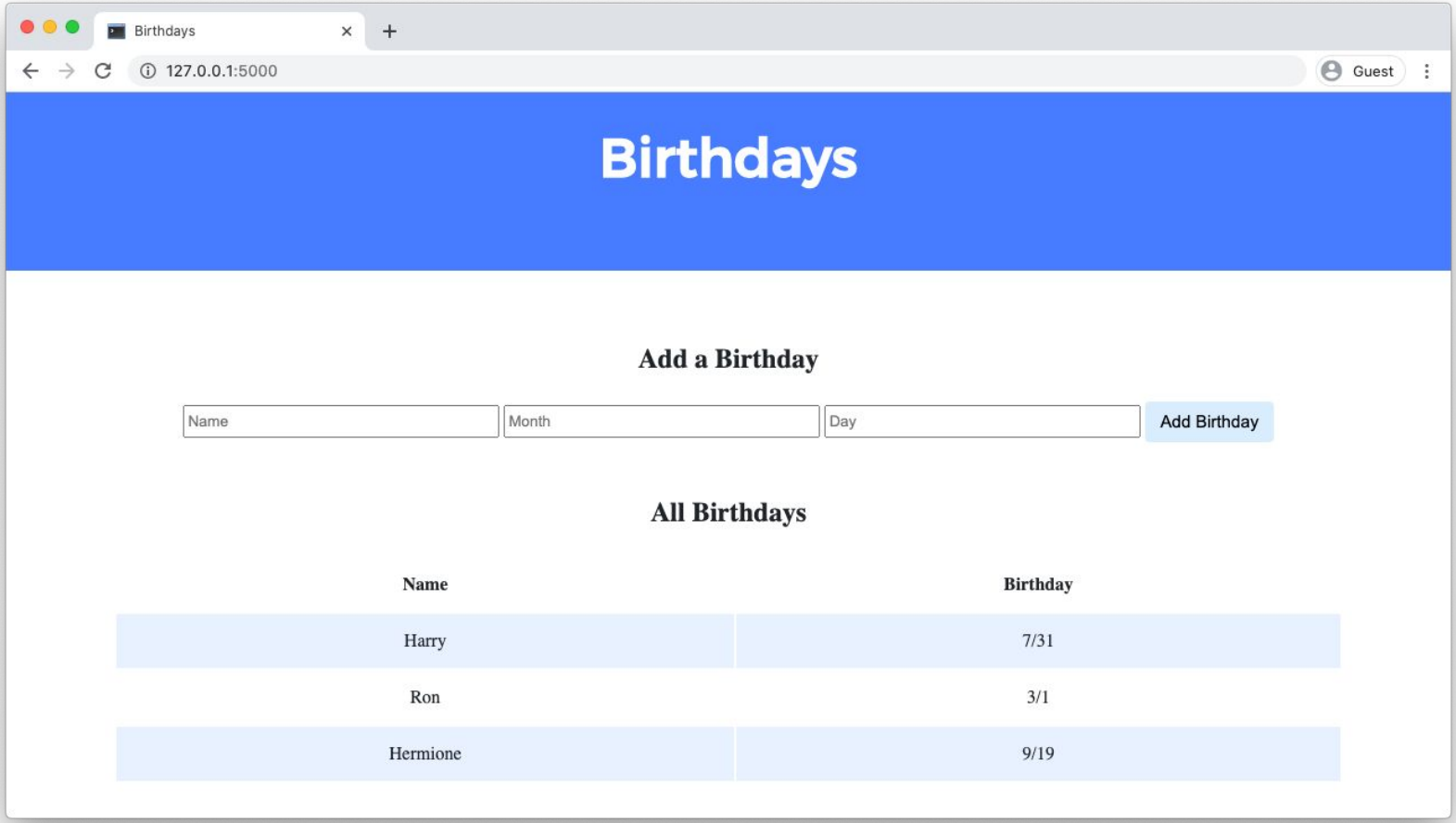
Project Due

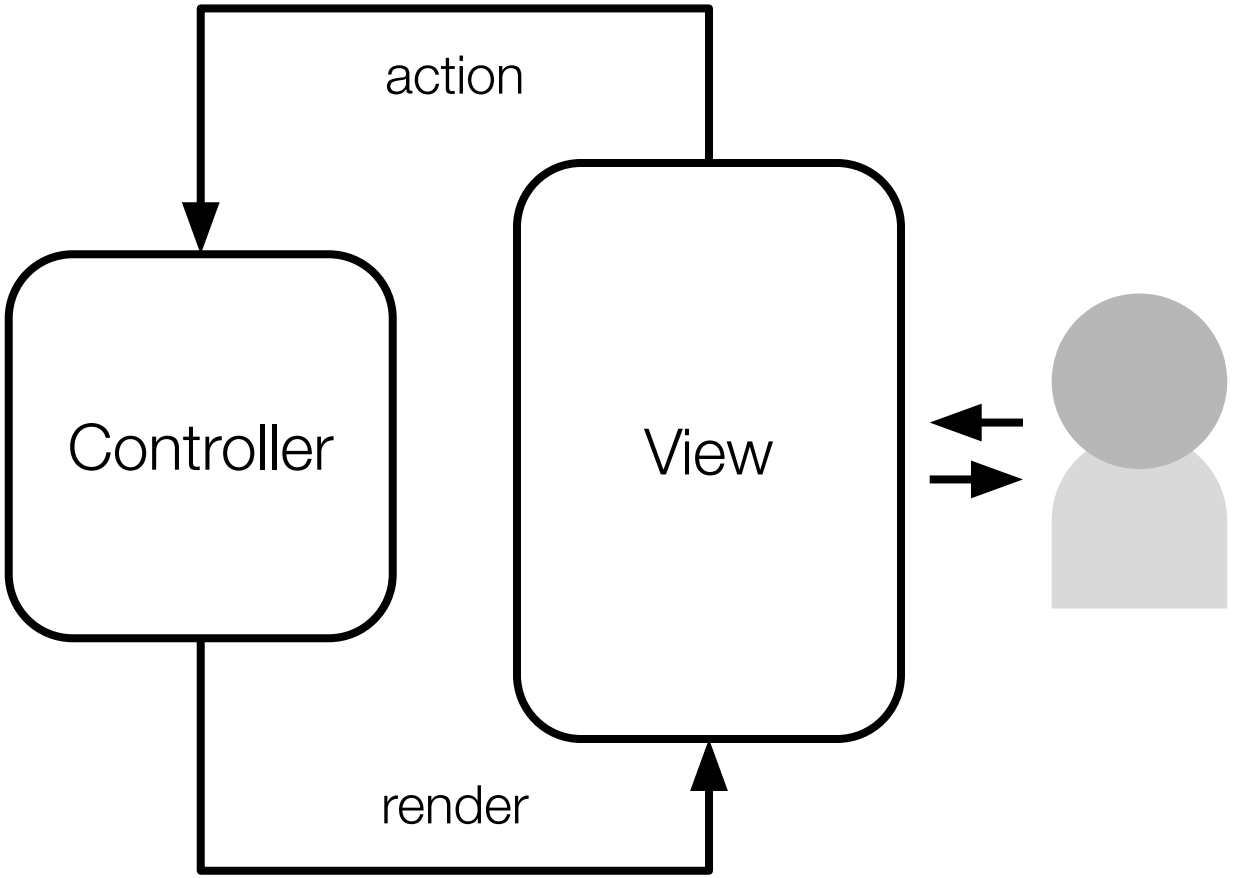
Think.
Pair.
Share.

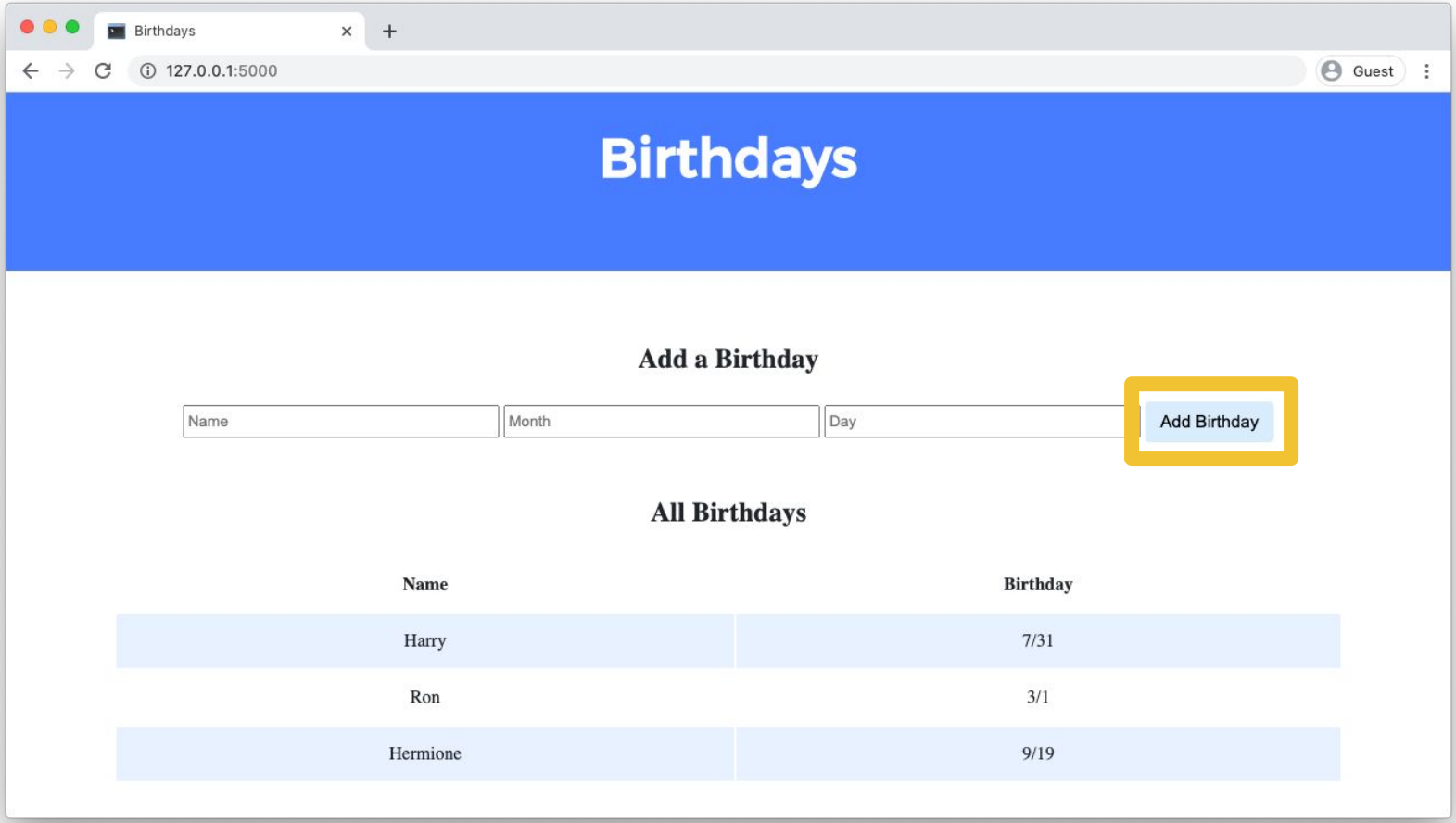
cs50.ly/questions

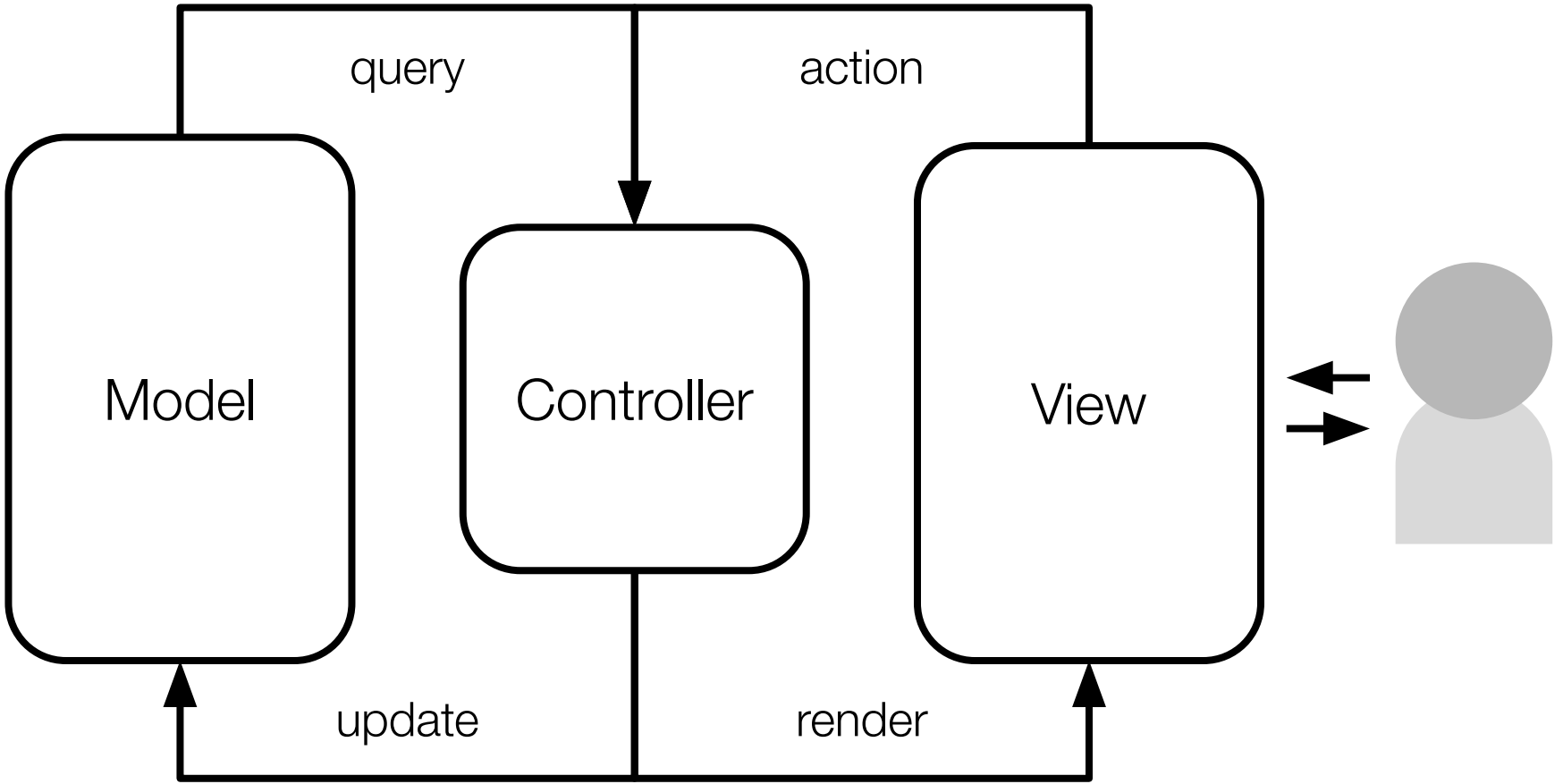

Birthdays
127.0.0.1:5000
Guest
Birthdays
Add a Birthday
Name
Month
Day
Add Birthday
All Birthdays
Name
Birthday
Harry
7/31
Ron
3/1
Hermione
9/19

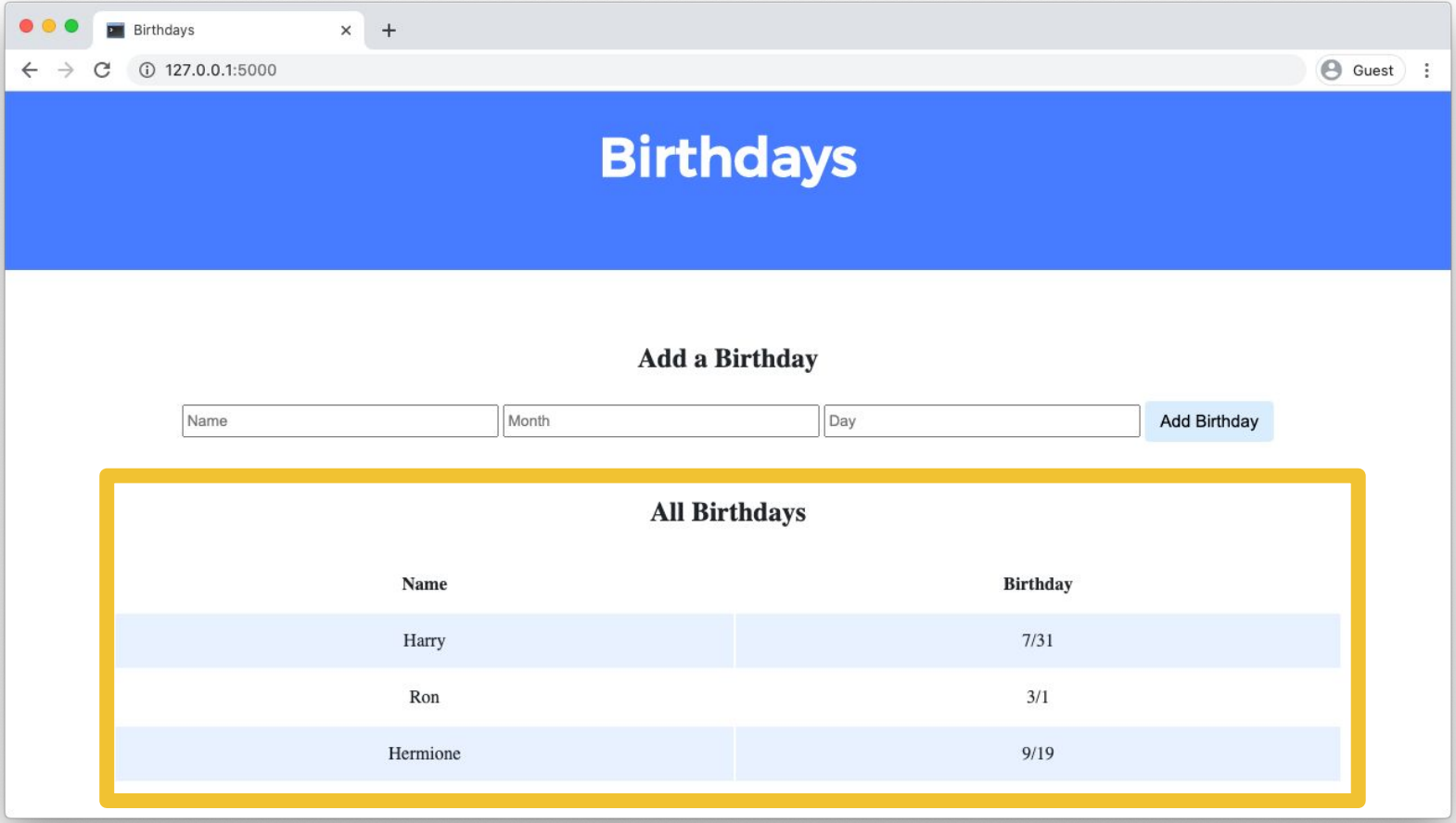

View


Birthdays
127.0.0.1:5000
Guest
Birthdays
Add a Birthday
Name
Month
Day
Add Birthday
All Birthdays
Name
Birthday
Harry
7/31
Ron
3/1
Hermione
9/19


action
Controller
View
render


Birthdays
127.0.0.1:5000
Guest
Birthdays
Add a Birthday
Name
Month
Day
Add Birthday
All Birthdays
Name
Birthday
Harry
7/31
Ron
3/1
Hermione
9/19


query
action
Model
Controller
View
update
render


Birthdays
127.0.0.1:5000
Guest
Birthdays
Add a Birthday
Name
Month
Day
Add Birthday
All Birthdays
Name
Birthday
Harry
7/31
Ron
3/1
Hermione
9/19

```
birthdays/
    static/
        styles.css
    templates/
        index.html
    app.py
    birthdays.db
```

Routes

https://birthdays.net/

https://birthdays.net/add

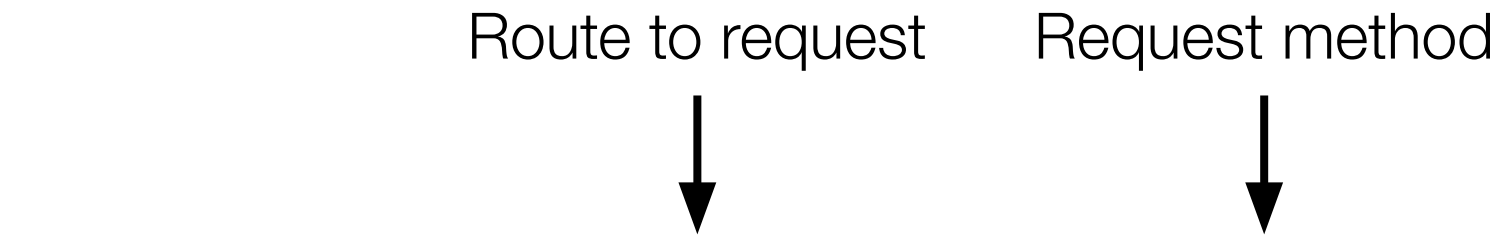
```
@app.route("/")
def index():
    # Return index page ("homepage")
```

```
@app.route("/")
def index():
    return render_template("index.html")
```

```
flask run
```

Forms

Form element



Elements of forms

```
<input name="friend" type="text">

<input name="month" type="number">

<button type="submit">Submit</button>
```

Elements of forms

```
<input name="friend" type="text">

<input name="month" type="number">

<button type="submit">Submit</button>
```

Submits form when clicked!

Add a form

In **index.html**, add a form to submit new birthdays. The form should have the following attributes:

- The form should submit to the default "/" route, using the POST method.
- The form should include an **input** for a **name**, as well as the **month** and **day** of a birthday.
- The form should include a button to **submit** the form.

Updating a model

```
request.form.get("friend")
```

```
<input name="friend" type="text">
```

```
request.args.get("friend")
```

```
<input name="friend" type="text">
```

```
INSERT INTO table (column1, column2)
VALUES (value1, value2);
```

```
db.execute("")
```

```
db.execute("?", value1)
```

```
db.execute("? ?", value1, value2)
```

Update a database

In **app.py**, insert a new entry to the **birthdays** table of **birthdays.db** when the user submits data via POST:

- Use **request.form.get("")** to get the values from each named input element.
- Use **db.execute("")** with **?** placeholders to insert those values into the **birthdays** table.
- Be sure to *validate* user input. For example, check if a value is **None** before inserting the value into the database.

Template Rendering

```
render_template("index.html",
                message="Hello")
```

```
<p>The message is: {{ message }}</p>
```

```
render_template("index.html",
                bday_list=["Feb 2", "Feb 28"])
```

```
{% for bday in bday_list %}
    <p>{{ bday }}</p>
{% endfor %}
```

```
render_template("index.html",
                bday={"month":"Feb", "day":24})
```

```
<p>This birthday is in {{ bday.month }}</p>
```

`{% ... %}` Jinja expressions (e.g., **for**)

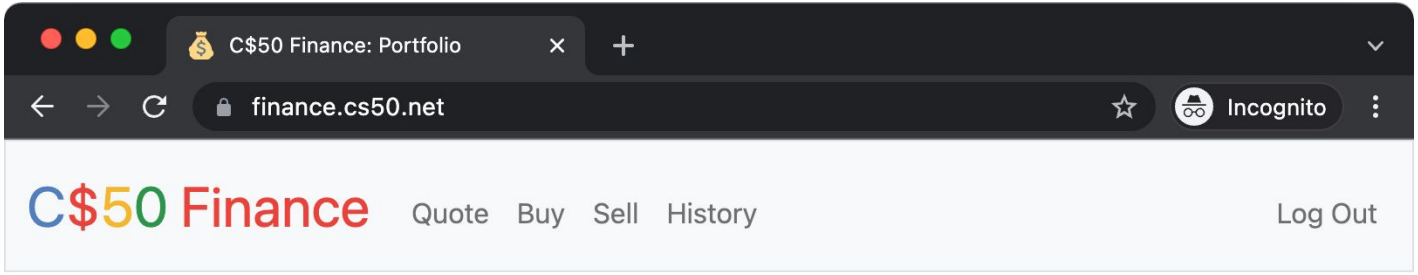
`{{ ... }}` Jinja variables (e.g., **bday**, **message**)

Render birthdays

In **app.py** and **index.html**, query for birthdays and display them in a table.

- Use **db.execute("")** to query for a list of birthdays in the **birthdays** table.
- Use Jinja syntax to print out birthdays one by one in a table.

Problem Set 9



Symbol	Name	Shares	Price	TOTAL
NFLX	NetFlix Inc	1	$655.63	$655.63
			Cash	$9,344.26
			TOTAL	$9,999.89

Data provided by IEX

Problem Set 9 Tips

- Start early
- Recall that **db.execute("SQL Query")** will return a Python list of dictionaries, where each dictionary is a row with keys corresponding to column names.
- When designing your database, try to **minimize redundancy**. If you already have a user's history of transactions, how could you derive their portfolio?
- Remember to **validate** user input before using it! Are values numbers where they should be? In the right range?
- A demo application is available https://finance.cs50.net/

Tutorials

Office Hours

cs50.ly/attend

This was CS50

This was CS50 :,)