User Accounts

No screens

Prof. Lydia Chilton
COMS 6998
5 October 2018
You already know **front-end** web dev:
HTML, JavaScript, Bootstrap, jQuery

And design:
Iterative design, critique

You will learn **back-end** web dev:
- Server-side programming (Flask),
- **Databases** (Sqlite, SQLAlchemy)
- Real-time Communication (Socket.IO)

And practice web design by:
- **Rebuilding** IMDB.com
- Rebuilding twitter
- Pursuing your own project
Last week:

Build a web app that meets a specific user need in the movie data domain

(That is not a specific need that IMDB supports well)
Last week’s goal: Iterate

• Iterate on your application
  • Define your own high level goal: a specific user need to solve
    • Define 7-10 low level goals you will need to achieve this.
  • This should be the next riskiest thing

• It should now be feature complete and ready for users!
  • Give a user persona so you know that it is ready for at least one user.
  • The graphic design should reflect the intent of the application to the user.
    • It can be minimal, but it should convey the purpose of the side and guide the user’s attention through the task
  • Create at least 1 paper prototype – test it on at least 1 person.
    • Report one thing you learned from it
  • Test it on at least 1 person
    • Report on thing you learned
Use graphic design to convey the specific needs your site serves.

- Conceptual grouping
- Location
- Whitespace
- Size
- Images
- Contrast
- Color
Book unique homes and experiences.

WHERE
Anywhere

CHECK IN
mm/dd/yyyy

CHECK OUT
mm/dd/yyyy

GUESTS
1 guest

Search
### Course Listings by Subject

| Course Listings by Subject | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

### Course Listings by Department

| Course Listings by Department | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

### Keyword Search

- **Keywords:**
- **Instructor:**
- **Semester:**
- **Weekday:**
- **Class start time:**

### Affiliated Institutions

- Teachers College course information

### Home Pages

- Instructors with home pages

---

*NOTE: Course information changes frequently. Please re-visit these pages periodically for the most recent and up-to-date information.*
Studio Time: 45 minutes

Critique:

• What specific user need is being met?

• Does the graphic design help the user meet their specific need in an intuitive way?
How to give

Negative Critique
To genuinely help people we can’t be too nice. We have to say the things that aren’t working.
I asked some of you: 
“Is there are reason you’re not as providing incisive negative critique?”

I don’t want to risk being too mean to somebody’s face.
Interface #1

• Specific User need:
  • Help advanced language learners immerse themselves in the language and culture of that language by finding films in their target language.
Immerse yourself. Watch a movie in Danish.

Improve your language skills and expand your cultural frontiers
Studies show that watching movies is one of the best ways to get better at language and achieve native-level proficiency. Relax and soak up new worlds of cinema you didn’t know existed, whether you’re hanging out with friends or just chilling in bed.

How it works
We make it convenient for you to discover great movies in the language you’re learning. By aggregating movies using information from IMDb and other sources, we provide you with movies you’ll be interested in that enrich your linguistic knowledge.
### Find your language

- Danish
- English
- Spanish
- Finnish
- Irish
- Kurdish
- Cornish
- Luxembourgish
- Polish
- Swedish
- Turkish
- Yiddish
Movies in Japanese

Classics

- Spirited Away
  2003 - 8.6/10
- Princess Mononoke
  1997 - 8.4/10
- Seven Samurai
  1956 - 8.7/10
- Howl's Moving Castle
  2005 - 8.2/10
- My Neighbor Totoro
  1988 - 8.2/10

New releases

- Junpei, Think Again
  2018
- Before The Coffee Gets Cold
  2018
- Hibiki - shōsetsuka ni naru hōhō
  2018
- Real Girl
  2018
- Okko's Inn
  2018
with her companion Dodi Al Fayed...

Now the cheese shop.
Picodons are 12.90, cabecous 23.50.

All women want to sleep on a man's shoulder.

- Nobody's immune.
- At least it gives me a break.

At the same time, a man leaves his home at 108 Rue Lecour

- Did it work?
- It's getting there.

I saw their little game.
Critique: I like, For me.../ I wish...

<table>
<thead>
<tr>
<th>Classics</th>
<th>New releases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spirited Away</strong> 2001 - 8.6/10</td>
<td><strong>Junpei, Think Again</strong> 2018</td>
</tr>
<tr>
<td><strong>Princess Mononoke</strong> 1997 - 8.4/10</td>
<td><strong>Before The Coffee Gets Cold</strong> 2018</td>
</tr>
<tr>
<td><strong>Seven Samurai</strong> 1954 - 6.7/10</td>
<td><strong>Hibiki - shōsetsuka ni naru hôhô</strong> 2018</td>
</tr>
<tr>
<td><strong>Howl's Moving Castle</strong> 2005 - 8.2/10</td>
<td><strong>Real Girl</strong> 2018</td>
</tr>
<tr>
<td><strong>My Neighbor Totoro</strong> 1988 - 8.2/10</td>
<td><strong>Okko's Inn</strong> 2018</td>
</tr>
</tbody>
</table>
Iteration:
Made the Movies more like Netflix
Interface #2

• Specific User need:
  • “I wanted to practice using a python backend to call a Markov Chain, so I used to it generate movie titles. So it can help script writer come up with ideas, but sometimes it’s just funny.”
The next great Western film...
Track of the Bad Guys.

Revenge Mystery Danger Ambush It Will Pleasure You in a fight for GOLD and a beautiful background.
The next great Comedy film...
A Walk in the Attic.

When life falls apart, friends keep it together. Opportunity rocks where you start - It's the Comedy Smash of the expected!
The Treasure of the Mohicans.

The most daring rescue mission of life and running from people who should know better.
Critique: I like..., For me... I wish...
Studio Time: 45 minutes

Critique:
- Present your specific needs
- Present your interface by walking us through a user’s path through the system to meet their goal.
- Everyone presents, everyone critiques.
  - What specific user need is being met?
  - Does the graphic design help the user meet their specific need in an intuitive way?

Today by 9pm – How will you iterate on your graphic design
AND
what is (at least) one feature will you add that requires user accounts
User Accounts
Basic elements of Login

Front end:
User actions needed

Register
New Users

Login

Logout

Using app

Back end:
Stuff needed to support user actions

Database of users

Password hashing

Register

Bad: User already exists

Correct

Good: created. Now what?

Incorrect

Where to send the user next

Know they are signed in
What to implement:

**Register New Users**

**Login**

**Logout**

**Using app**

**Front end: User interface**
- A form
- feedback

**Back end: Database interaction**
- Validation

Reminder you are logged in

nothing
### What to implement:

<table>
<thead>
<tr>
<th></th>
<th>Front end: User interface</th>
<th>Back end: Database interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register New Users</td>
<td>A form feedback</td>
<td>Validation</td>
</tr>
<tr>
<td>Login</td>
<td>A form feedback</td>
<td>Validation</td>
</tr>
<tr>
<td>Logout</td>
<td>A button</td>
<td>nothing</td>
</tr>
<tr>
<td>Using app</td>
<td>Reminder you are logged in</td>
<td>nothing</td>
</tr>
</tbody>
</table>

**Front end:**
- User interface
- Database interaction

**Back end:**
- Validation

**Frameworks:**
- WTForms
- Flask_login

**Notes:**
- Using app: Reminder you are logged in
- Front end: User interface
- Back end: Database interaction
**Flask_login:**
log users in, log users out, know who they are

**Add flask_login:**

```python
from flask_login import LoginManager

login = LoginManager(app)

@login.user_loader
def load_user(id):
    return User.query.get(int(id))
```

**Use flask_login:**

```python
user = Users.query.get(id)

login_user(user)

<div>
    Hi, {{ current_user.username }}
</div>
```
WTForms

Define forms and their validation

**Back-end:**
Define Form class

**Front end:**
render form

```python
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, BooleanField,
from wtforms.validators import ValidationError, DataRequired,
from app.models import User

class LoginForm(FlaskForm):
    username = StringField('Username', validators=[DataRequired()])
    password = PasswordField('Password', validators=[DataRequired()])
    remember_me = BooleanField('Remember Me')
    submit = SubmitField('Sign In')
```
This week’s goal: Implement **User Accounts**

- Extend your application by adding one feature that requires user accounts.
  - Define your own high level goal
    - Define 7-10 low level goals you will need to achieve this.
  - You must implement **user accounts**
  - **Add a feature** to your site that requires user account.
  - The **graphic design** should reflect the intent of the application to the user.
    - Please iterate on the graphic design based on your feedback today.

Today by 9pm – How will you iterate on your graphic design **AND** what is (at least) one feature will you add that requires user accounts