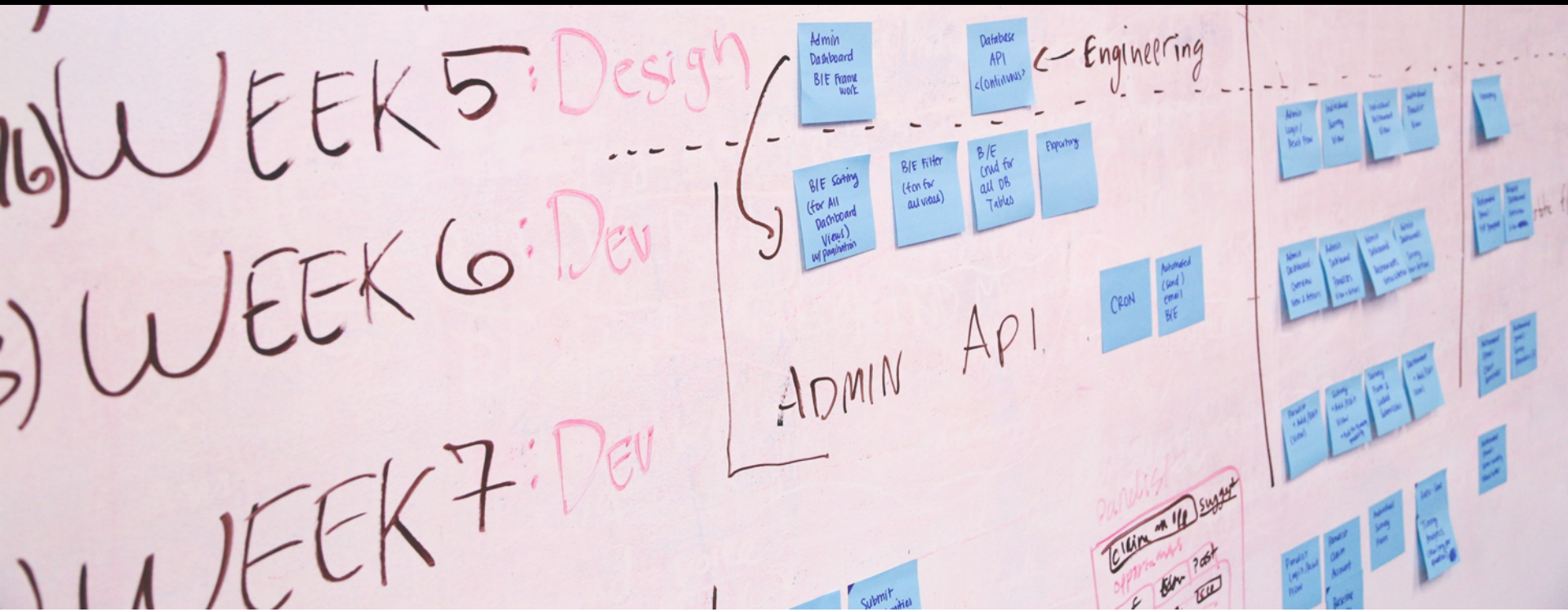


COM5961 DATA DRIVEN PRODUCTS & SERVICES DESIGN:
LESSON 8 - PYTHON WEB PROGRAMMING IN FLASK II

Bernard Suen
Center for Entrepreneurship
Chinese University of Hong Kong

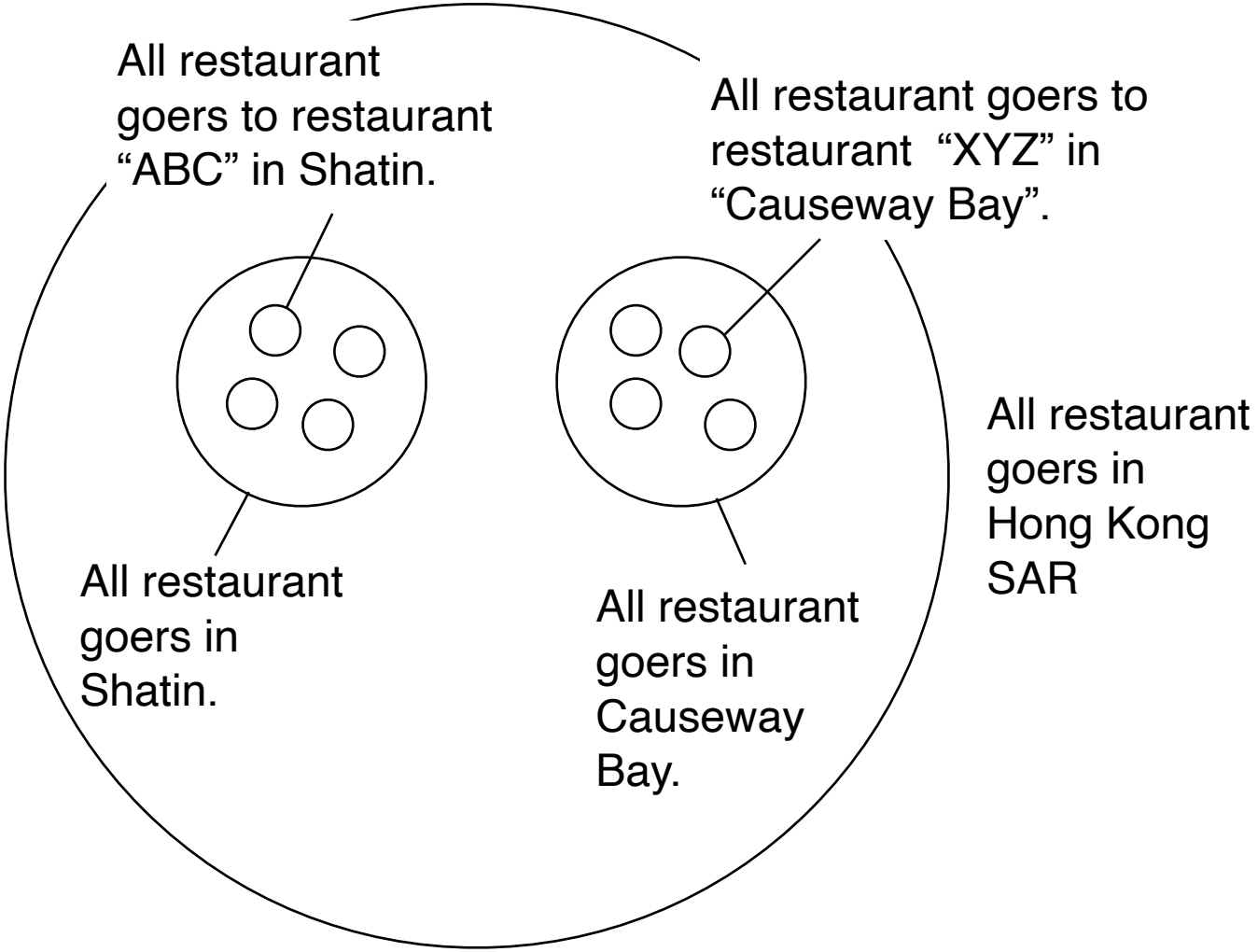
Today's agenda.

1. Quick comments on assignment #2.
2. From local development on Jupyter Notebook to remote deployment on PythonAnywhere cloud service.
3. Integrating SQLite data into Flask HTML templates.
4. Usability tests for future UX hypothesis validation using Figma wire frames.
5. Transform wireframes into Flask HTML templates for deployment testing.

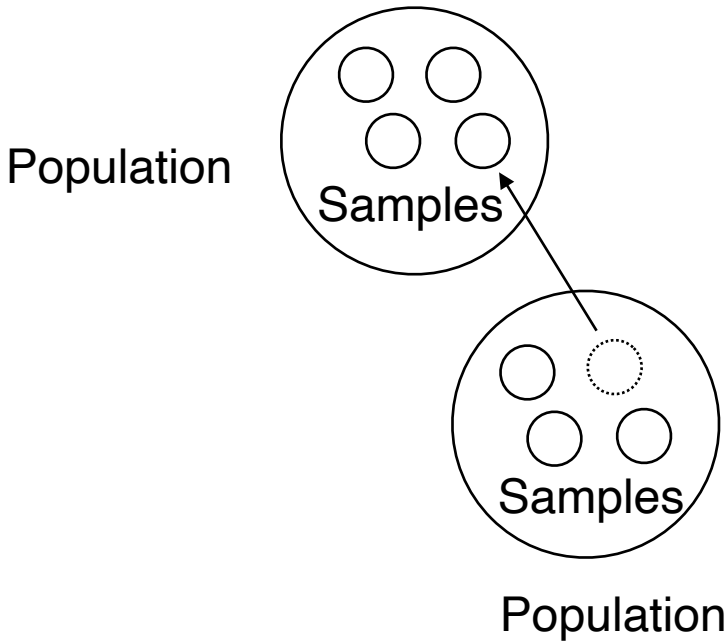
Quick comments on assignment #2.

From descriptive statistic to inference statistic: the missing link.

Can a person who visited "ABC" also visited "XYZ"?



Concept of “sample” and “population”.



How can we tell that the members are truly from the same group but not my chance?

Don’t over generalise without enough factual evidence.

SQL is a Great Tool for Data Analysis

via Sports Logos.net
About logos

2014 NBA Draft

« 2013 NBA Draft 2015 NBA Draft »

Date: Thursday, June 26, 2014
Location: New York, New York
Number of Picks: 60 (53 played in NBA)
First Overall Pick: Andrew Wiggins (24.0 Win Shares)
Most Win Shares: N. Jokić (79.6), C. Capela (52.9) and J. Embiid (43.7)
All-Stars: 5 (J. Embiid, N. Jokić, Z. LaVine, J. Randle and A. Wiggins)

	Col_Player	Col_College	Col_Game	Col_Yrs	Games/Years	Points	Avg Poits/Yr
0	Zach LaVine	UCLA	478	8	59.75	9466	158.43
1	Joel Embiid	Kansas	328	6	54.67	8535	156.13
2	Andrew Wiggins	Kansas	598	8	74.75	11519	154.10
3	Julius Randle	Kentucky	518	8	64.75	9191	141.95
4	Jordan Clarkson	Missouri	600	8	75.00	9216	122.88
5	Jabari Parker	Duke	310	8	38.75	4380	113.03
6	T.J. Warren	NC State	332	7	47.43	5142	108.42
7	Aaron Gordon	Arizona	528	8	66.00	6887	104.35
8	Spencer Dinwiddie	Colorado	387	8	48.38	5042	104.23
9	Gary Harris	Michigan State	468	8	58.50	5526	94.46
10	Jerami Grant	Syracuse	555	8	69.38	6329	91.23

	Col_College	Total Avg Poits/Yr
0	Kansas	310.23
1	UCLA	218.06
2	Duke	196.17
3	Kentucky	151.17
4	Missouri	122.88
5	Michigan State	110.50
6	NC State	108.42
7	Syracuse	107.98
8	Arizona	106.99
9	Colorado	104.23
10	Oklahoma State	99.16

October 8, 2022 **David Doe**

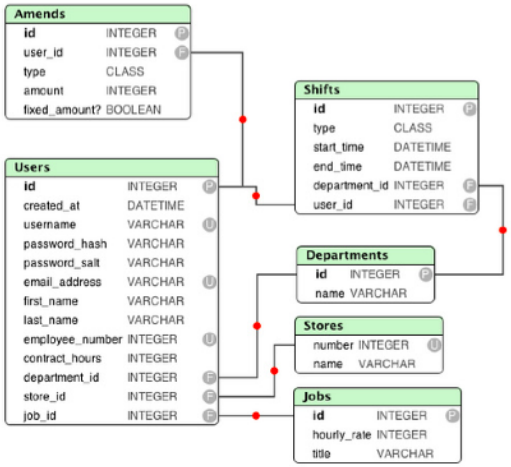
Scraping is a useful source for obtaining data among the various methods that are available out there. Nevertheless, the work does not stop there as the scraped data often needs to be cleaned and transformed into the proper format and structure until it can be further used.

In the 2nd assignment, ParseHub and Open Refine were used to help get the data into a usable state by removing blank and null entries for performing further data analysis. The problem at hand was: How can I find out the top 10 players and colleges with the highest average score per year from the NBA 2014 Draft

Python Web Development in Flask: Transforming HTML files to Flask HTML templates.

Source: commons.wikimedia.org

Source: Caius Durling (Flickr)



M(odel)
(SQL/NoSQL data models and processing functions)

V(iew)
(Flask templates)

Routing control to pre-defined data and functions

Source: pexels.com

C(ontroller)
(e.g. Routes, rights and authentication)

```
In [*]:
```

```
##########################################################################################################

# EXTERNAL MODULES TO BE USED

##########################################################################################################

from flask import Flask, render_template, redirect, url_for

app = Flask(__name__)

##########################################################################################################

# WEB ROUTES FOR CONTROLLING ACCESS TO TEMPLATE VIEWS

##########################################################################################################

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

@app.route("/assign1")
def assign1():
    return render_template('index.html')

@app.route("/assign2")
def assign2():
    return render_template('assign2.html')

@app.route("/assign3")
def assign3():
    return render_template('assign3.html')

@app.route("/assign4")
def assign4():
    return render_template('assign4.html')

@app.route("/assign5")
def assign5():
    return render_template('assign5.html')

##########################################################################################################

# ERROR HANDLERS

##########################################################################################################

@app.errorhandler(404)
def page_not_found(e):
    return render_template('404.html'), 404
```

1) Setting up the access route (e.g. @app.route(“assign1”).
2) Define the function associated with the route (e.g. def assign1()).
3) Return the HTML template used for displaying the web page (e.g. index.html).

```

##########################################################################################################

# APPLICATION TEST RUN AT PORT 9003

##########################################################################################################

if __name__ == '__main__':
    app.run('localhost', 9003)
```

Setting up the error handling route to handle “page not found”.
Define the error handling function associated with the route
Return the HTML template with the “Page not found” message on the web page.

jupyter index.html 20 hours ago

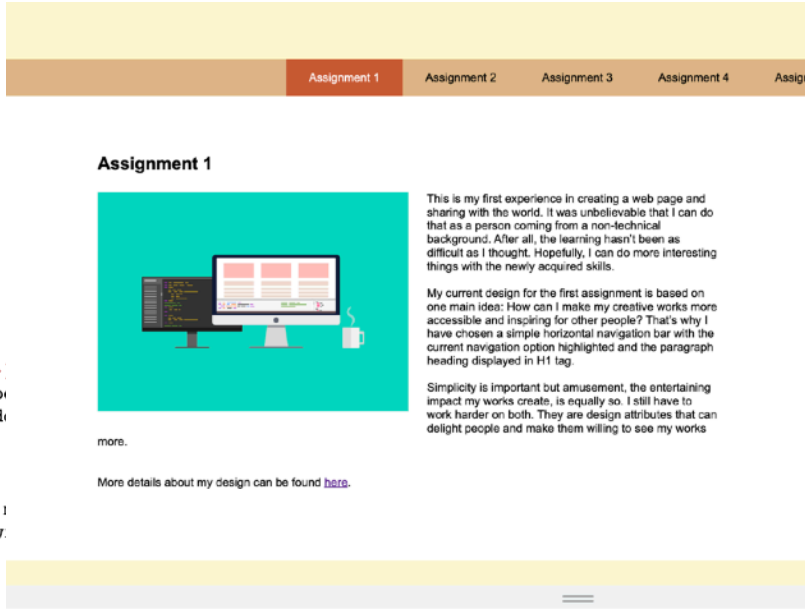
File Edit View Language

Flask HTML template

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="{{url_for('static', filename='css/style.css')}}">
</head>
<body>
<header><!--header section-->
<!-- <label id="menu" for="my_checkbox">Click For Menu</label> -->
<label id="menu" for="my_checkbox">
    <span class="bar"></span>
    <span class="bar"></span>
    <span class="bar"></span>
</label>
</header>
<input type="checkbox" id="my_checkbox">
<nav id="hidden">
    <ul>
        <li class="active"><a href="/assign1">Assignment 1</a></li>
        <li><a href="/assign2">Assignment 2</a></li>
        <li><a href="/assign3">Assignment 3</a></li>
        <li><a href="/assign4">Assignment 4</a></li>
        <li><a href="/assign5">Assignment 5</a></li>
    </ul>
</nav>
<main>
<section><!--main section-->
    <h1>Assignment 1</h1>
    <p>
        <img class="featured" src="{{url_for('static', filename='img/frontend-c3bacad8.png')}
creating a web page and sharing with the world. It was unbelievable that I can do that as a p
background. After all, the learning hasn't been as difficult as I thought. Hopefully, I can d
newly acquired skills.
    </p><br>
    <p>
        My current design for the first assignment is based on one main idea: How can I make
and inspiring for other people? That's why I have chosen a simple horizontal navigation bar w
highlighted and the paragraph heading displayed in H1 tag.
    </p><br>
    <p>
```

HTML File

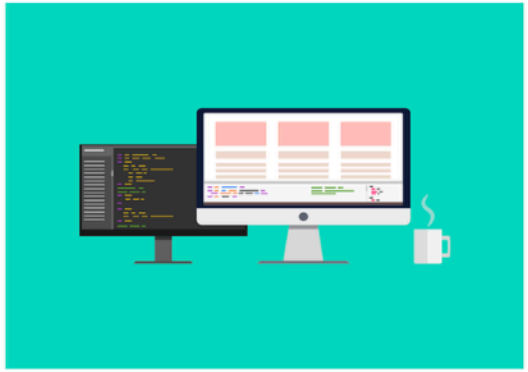
```
<header><!--header section-->
<!-- <label id="menu" for="my_checkbox">Click For Menu</lab
<label id="menu" for="my_checkbox">
    <span class="bar"></span>
    <span class="bar"></span>
    <span class="bar"></span>
</label>
</header>
<input type="checkbox" id="my_checkbox">
<nav id="hidden">
    <ul>
        <li class="active"><a href="index.html">Assignment
        <li><a href="assign2.html">Assignment 2</a></li>
        <li><a href="assign3.html">Assignment 3</a></li>
        <li><a href="assign4.html">Assignment 4</a></li>
        <li><a href="assign5.html">Assignment 5</a></li>
    </ul>
</nav>
```



```
if __name__ == '__main__':
    app.run('localhost', port=9000)
```

Running the application locally at a chosen port.

Assignment 1



This is my first experience in creating a web page and sharing with the world. It was unbelievable that I can do that as a person coming from a non-technical background. After all, the learning hasn't been as difficult as I thought. Hopefully, I can do more interesting things with the newly acquired skills.

My current design for the first assignment is based on one main idea: How can I make my creative works more accessible and inspiring for other people? That's why I have chosen a simple horizontal navigation bar with the current navigation option highlighted and the paragraph heading displayed in H1 tag.

Simplicity is important but amusement, the entertaining impact my works create, is equally so. I still have to work harder on both. They are design attributes that can delight people and make them willing to see my works more.

More details about my design can be found here.

My Journals

Making Remote Works Collaborative and Accountable

Rethink Digital Transformation

Lab time: Migrating your Python codes from Jupyter Notebook to PythonAnywhere.

Send feedback Forums Help Blog **Pricing & signup** Log in

pythonanywhere
by ANACONDA.

Host, run, and code Python in the cloud!

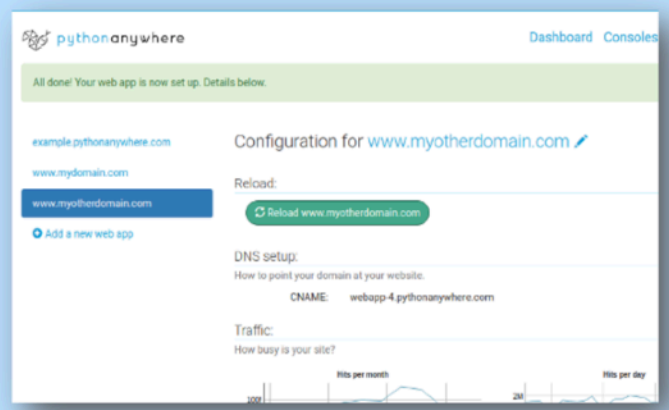
Get started for free. Our basic plan gives you access to machines with a full Python environment already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at $5/month.

Start running Python online in less than a minute! »

Watch our one-minute video »

Not convinced? Read what our users are saying!



Start hosting quickly

Just write your application. No need to configure or maintain a web server — everything is set up and ready to go.

More »

Develop anywhere

Take your development environment with you! If you have a browser and an Internet connection, you've got everything you need.

More »

Teach and learn

PythonAnywhere is a fully-fledged Python environment, ready to go, for students and teachers — concentrate on teaching, not on installation hassles.

Amazing support

Need help with PythonAnywhere? If you get in touch, you can talk directly with the development team. Help for developers, from developers.

More »

pythonanywhere by ANACONDA.

Plans and pricing

Beginner: Free!

A limited account with one web app at *your-username*.pythonanywhere.com, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython/Jupyter notebook support.
It works and it's a great way to get started!

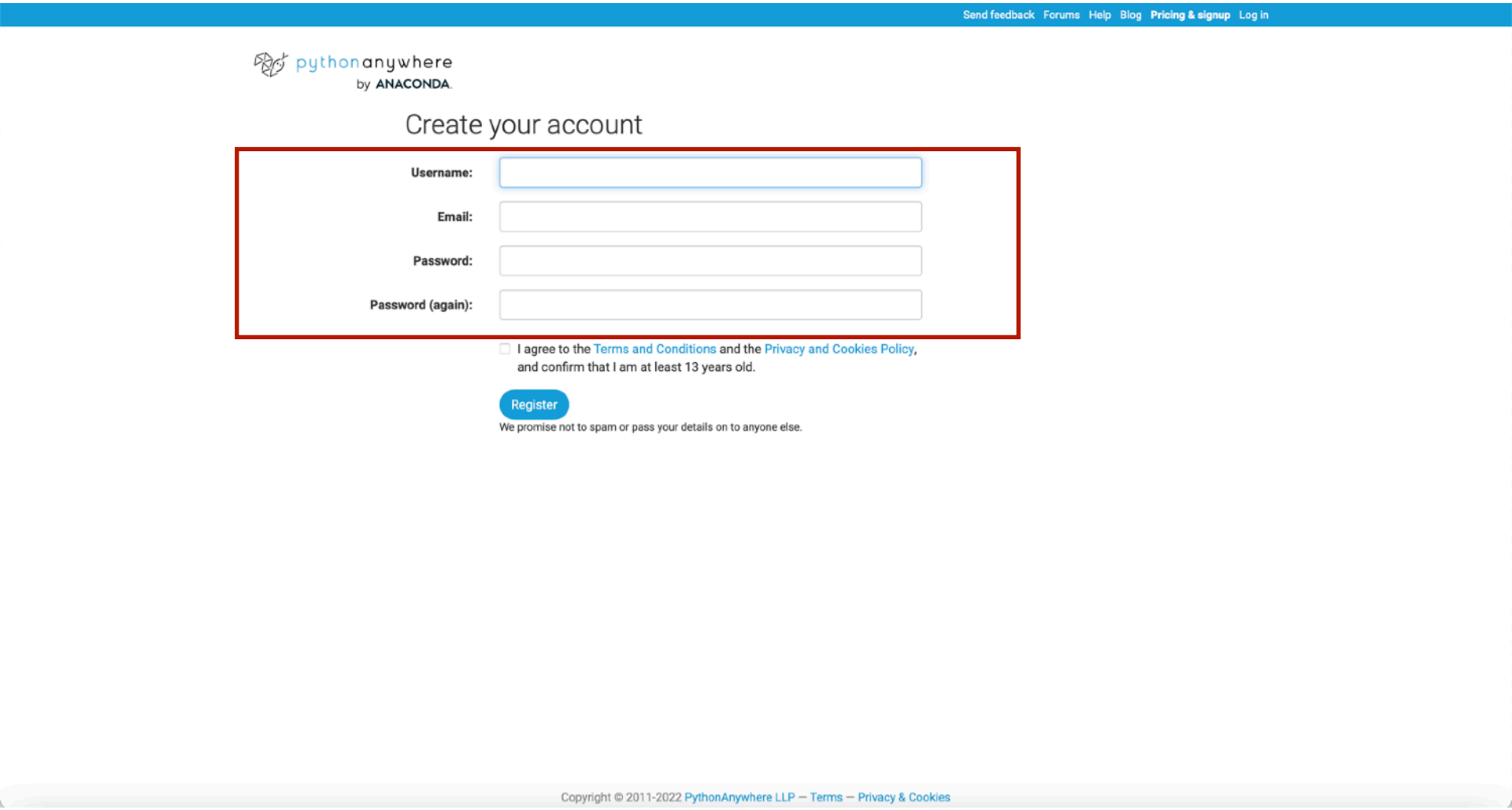
Create a Beginner account

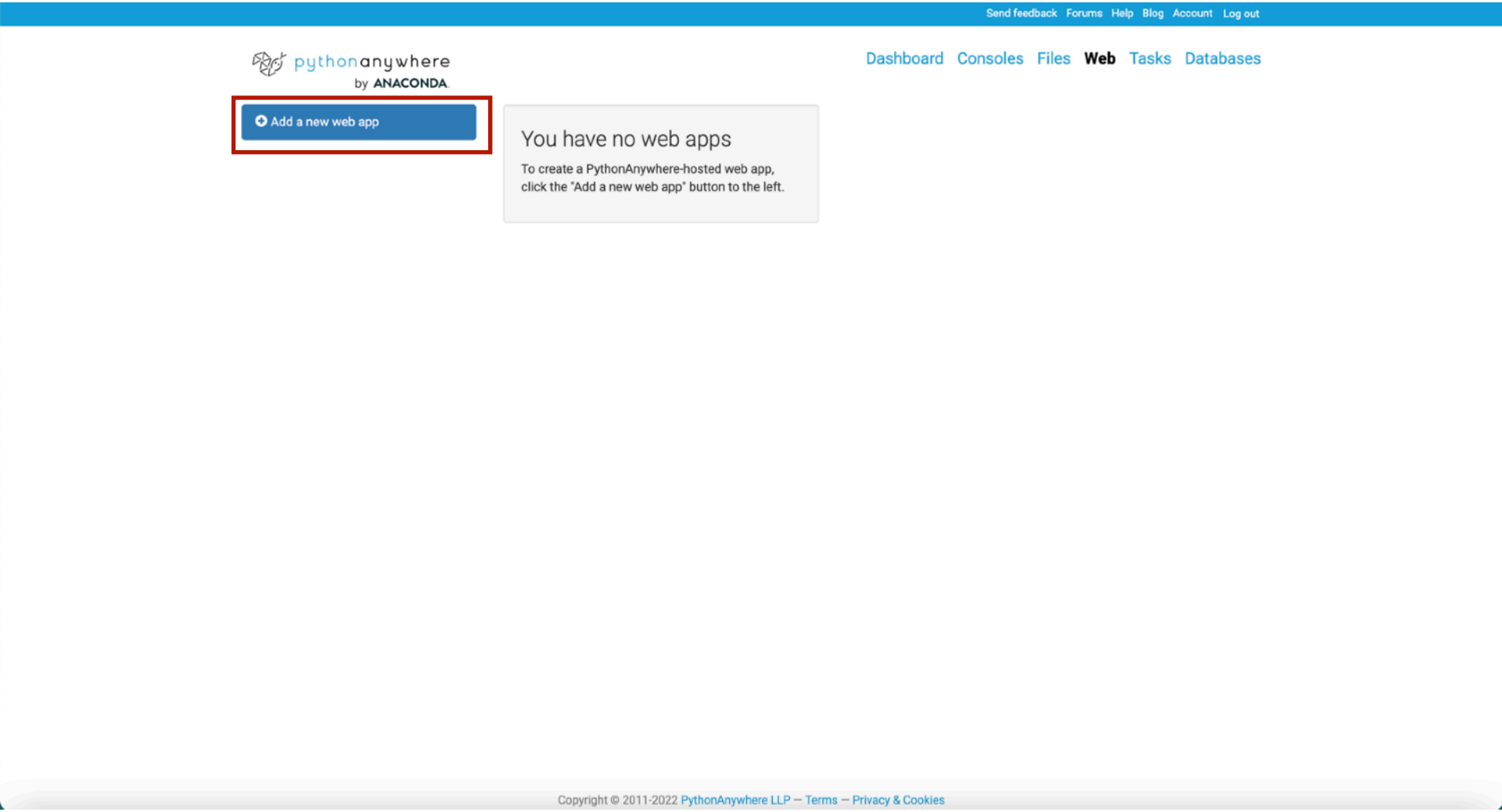
Education accounts

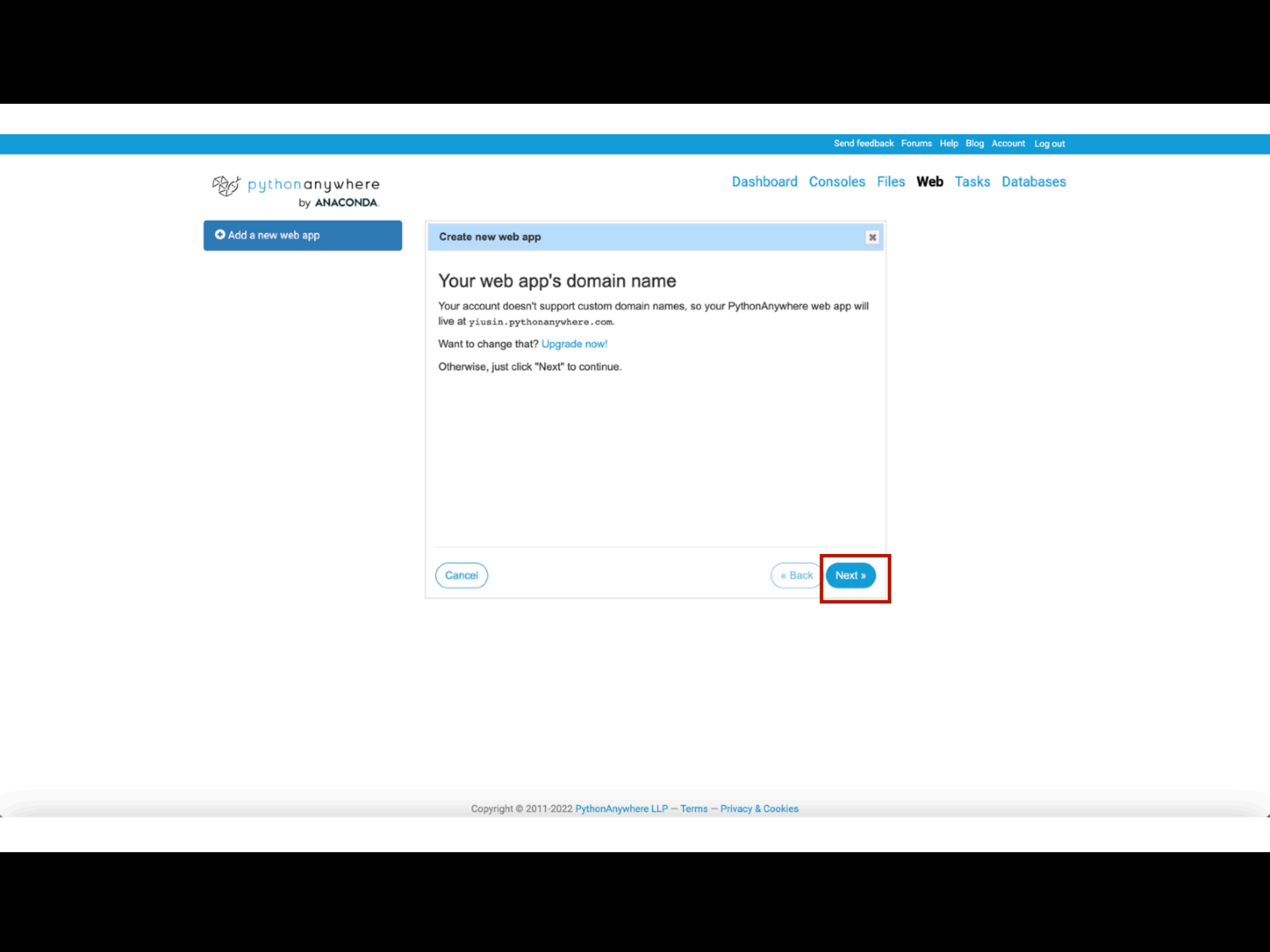
Are you a teacher looking for a place your students can code Python? You're not alone. Click through to find out more about our Education beta.

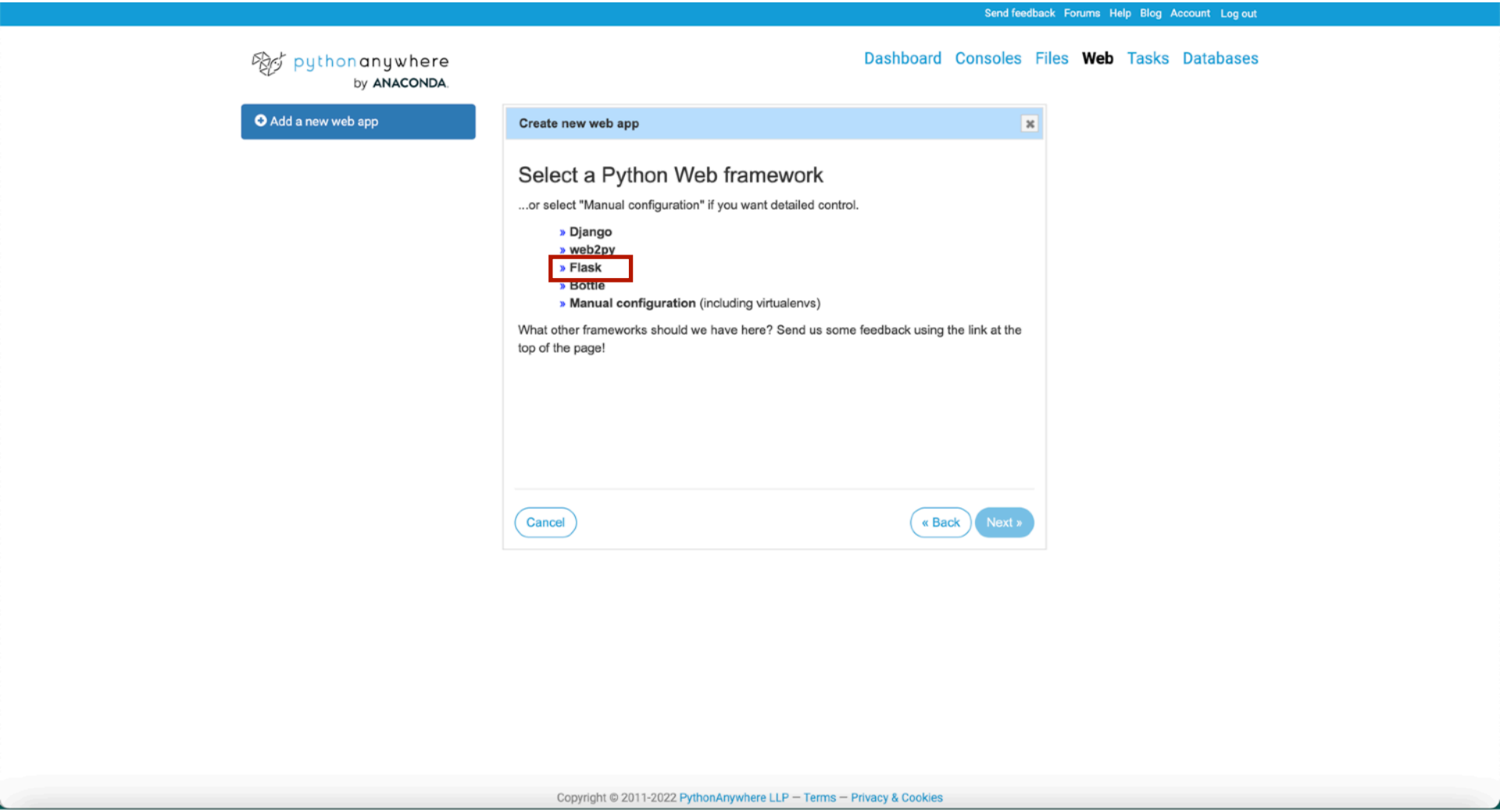
All of our paid plans come with a no-quibble 30-day money-back guarantee -- you're billed monthly and you can cancel at any time. The minimum contract length is just one month. You get unrestricted Internet access from your applications, unlimited in-browser Python, Bash and database consoles, and full SSH access to your account. All accounts (**including free ones**) have screen-sharing with other PythonAnywhere accounts, and free SSL support (though you'll need to get a certificate for your own domains).

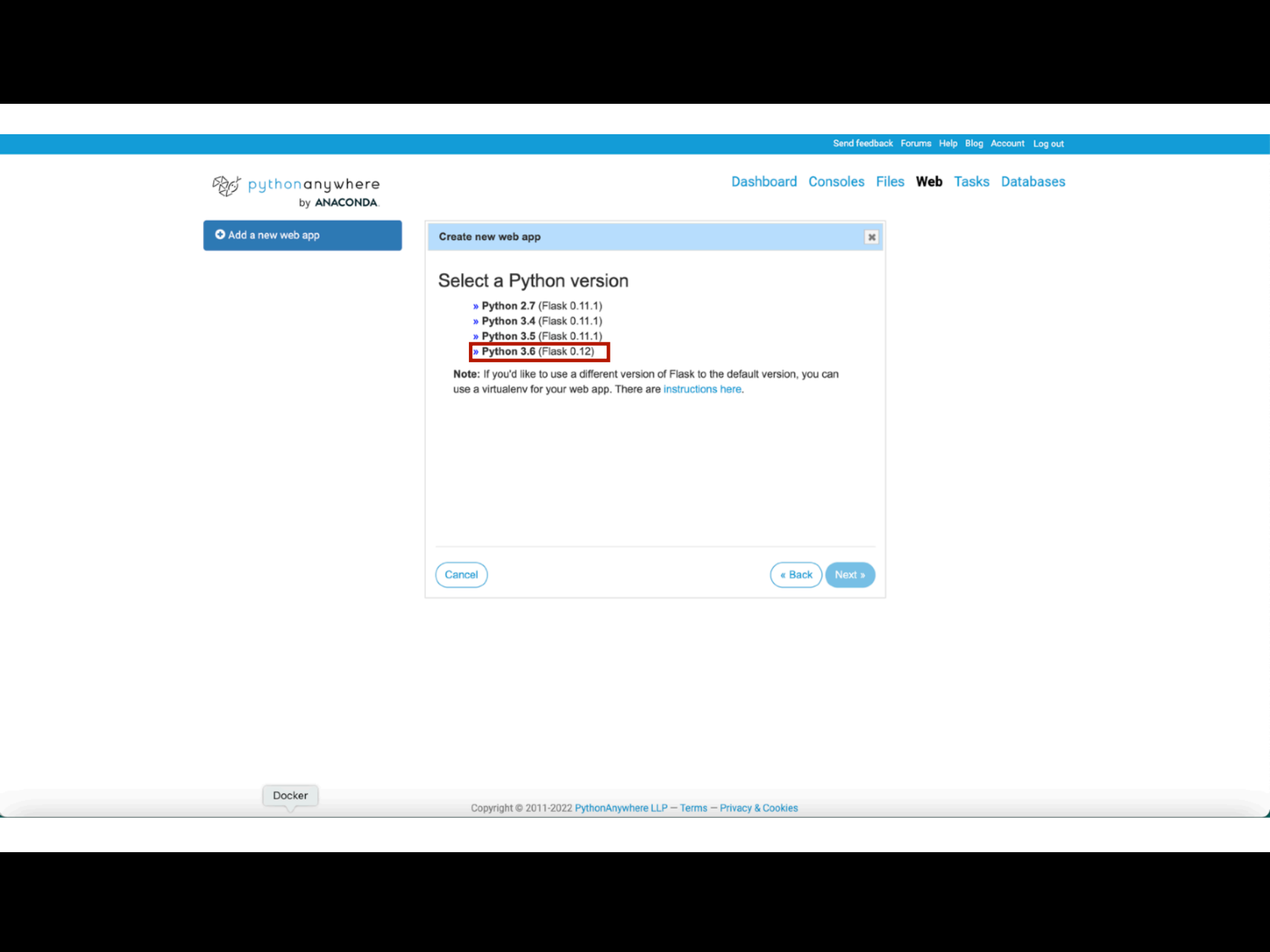
Hacker $5/month	Web dev $12/month	Startup $99/month	Custom $5 to $500/month
Run your Python code in the cloud from one web app and the console	**If you want to host small Python-based websites for you or for your clients**	**Start a business and don't worry about having to scale to handle traffic spikes**	**Want a combination that's not on the list? Create your own! All custom plans have:**
A Python IDE in your browser with unlimited Python/bash consoles	**A Python IDE in your browser** with unlimited Python/bash consoles	**A Python IDE in your browser** with unlimited Python/bash consoles	**A Python IDE in your browser** with unlimited Python/bash consoles
One web app on a custom domain or *your-username*.pythonanywhere.com	Up to **2 web apps** on custom domains or *your-username*.pythonanywhere.com	Up to **3 web apps** on custom domains or *your-username*.pythonanywhere.com	Up to **20 web apps**, on custom domains or *your-username*.pythonanywhere.com
Enough power to run a typical **100,000 hit/day website**. (more info)	Enough power to run a typical **150,000 hit/day website on each web app**. (more info)	Enough power to run a typical **1,000,000 hit/day website on each web app**. (more info)	**As many web workers as you need** to scale your site's capacity. (more info)
2,000 CPU-seconds per day for consoles, scheduled tasks and always-	4,000 CPU-seconds per day for		Up to 100,000 CPU-seconds per day


Send feedback Forums Help Blog Pricing & signup Log in
pythonanywhere
by ANACONDA.
Create your account
Username:
Email:
Password:
Password (again):
I agree to the Terms and Conditions and the Privacy and Cookies Policy, and confirm that I am at least 13 years old.
Register
We promise not to spam or pass your details on to anyone else.
Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies


Send feedback Forums Help Blog Account Log out
pythonanywhere
by ANACONDA.
Dashboard Consoles Files Web Tasks Databases
Add a new web app
You have no web apps
To create a PythonAnywhere-hosted web app, click the "Add a new web app" button to the left.
Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA.
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Create new web app
Your web app's domain name
Your account doesn't support custom domain names, so your PythonAnywhere web app will live at yiusin.pythonanywhere.com.
Want to change that? Upgrade now!
Otherwise, just click "Next" to continue.
Cancel
« Back
Next »
Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA.
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Create new web app
Select a Python Web framework
...or select "Manual configuration" if you want detailed control.
» Django
» web2py
» Flask
» Bottle
» Manual configuration (including virtualenvs)
What other frameworks should we have here? Send us some feedback using the link at the top of the page!
Cancel
« Back
Next »
Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA.
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Create new web app
Select a Python version
» Python 2.7 (Flask 0.11.1)
» Python 3.4 (Flask 0.11.1)
» Python 3.5 (Flask 0.11.1)
» Python 3.6 (Flask 0.12)
Note: If you'd like to use a different version of Flask to the default version, you can use a virtualenv for your web app. There are instructions here.
Cancel
« Back
Next »
Docker
Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies

Send feedback Forums Help Blog Account Log out

pythonanywhere by ANACONDA.

Dashboard Consoles Files **Web** Tasks Databases

Add a new web app

Create new web app

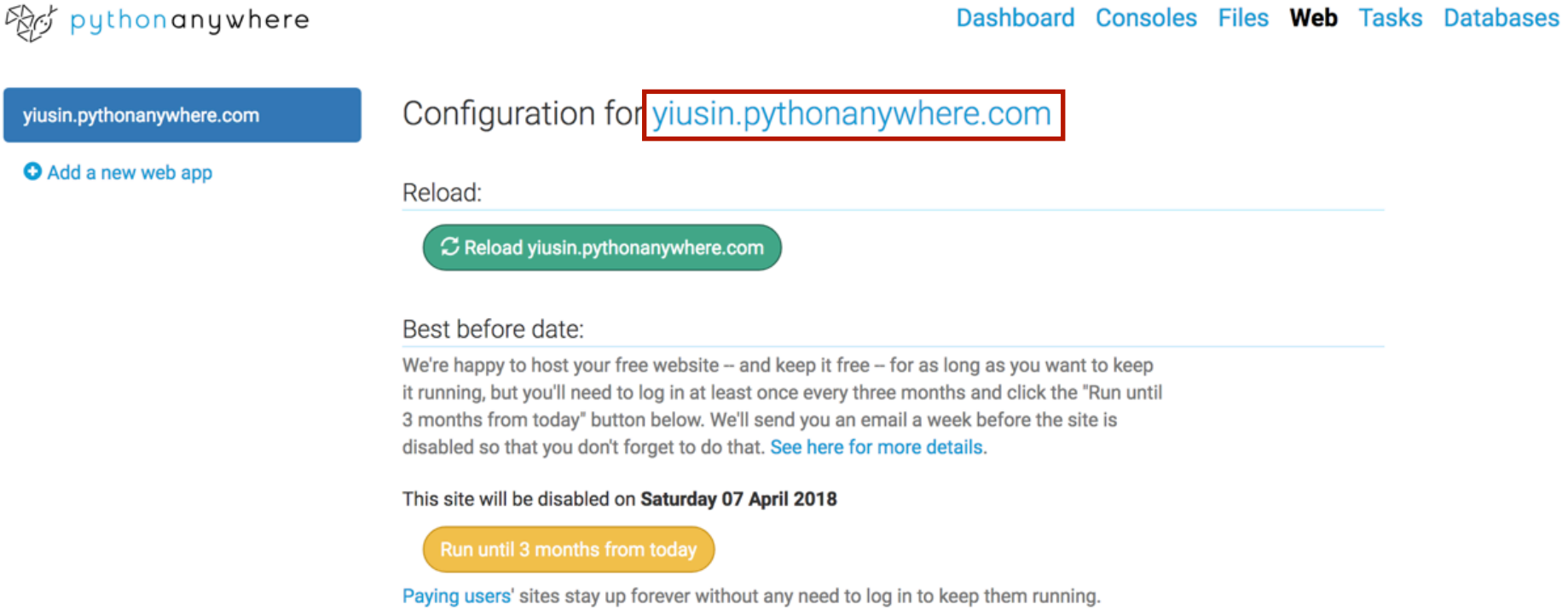
Quickstart new Flask project

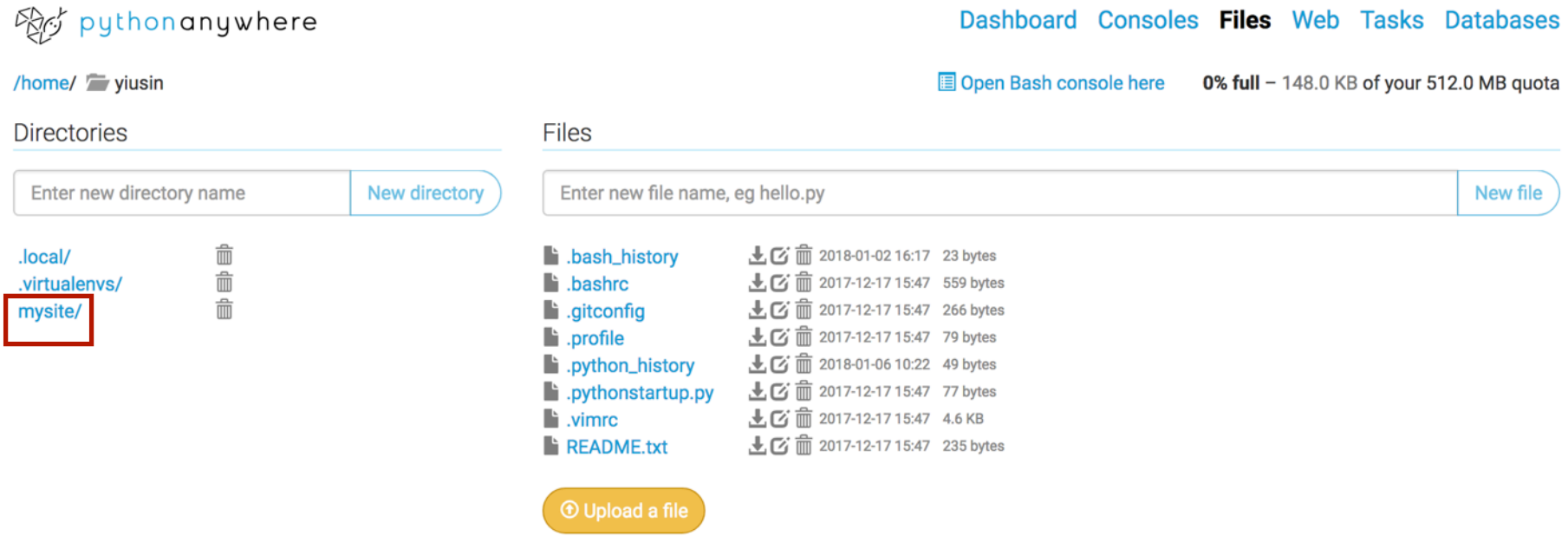
Enter a path for a Python file you wish to use to hold your Flask app. **If this file already exists, its contents will be overwritten with the new app.**

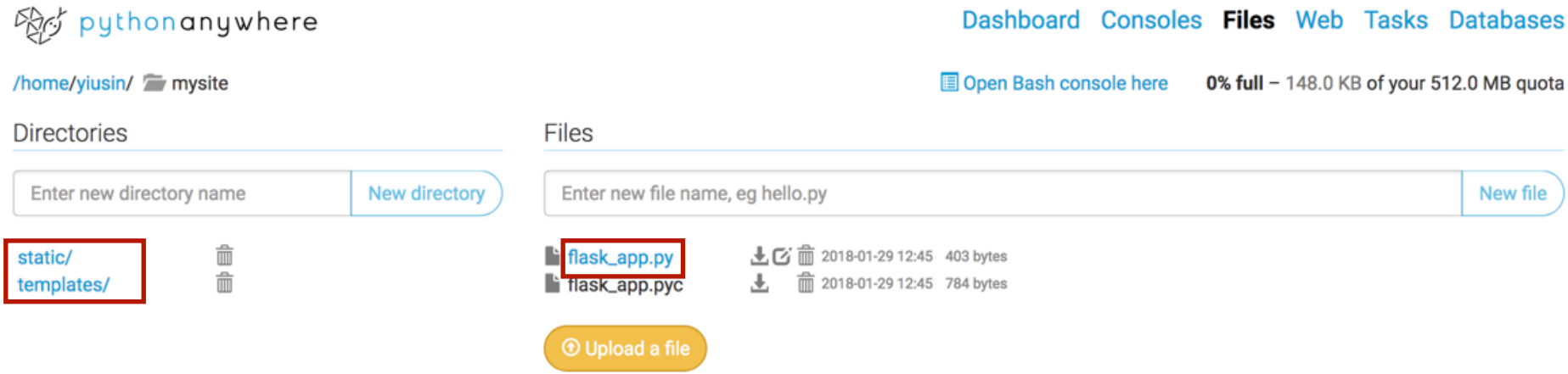
Path

/home/yiusin/mysite/flask_app.py

Cancel « Back Next »


pythonanywhere
Dashboard Consoles Files Web Tasks Databases
yiusin.pythonanywhere.com
Add a new web app
Configuration for yiusin.pythonanywhere.com
Reload:
Reload yiusin.pythonanywhere.com
Best before date:
We're happy to host your free website -- and keep it free -- for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. See here for more details.
This site will be disabled on Saturday 07 April 2018
Run until 3 months from today
Paying users' sites stay up forever without any need to log in to keep them running.


pythonanywhere
Dashboard Consoles Files Web Tasks Databases
/home/ yiusin
Open Bash console here
0% full – 148.0 KB of your 512.0 MB quota
Directories
Enter new directory name
New directory
.local/
.virtualenvs/
mysite/
Files
Enter new file name, eg hello.py
New file
.bash_history 2018-01-02 16:17 23 bytes
.bashrc 2017-12-17 15:47 559 bytes
.gitconfig 2017-12-17 15:47 266 bytes
.profile 2017-12-17 15:47 79 bytes
.python_history 2018-01-06 10:22 49 bytes
.pythonstartup.py 2017-12-17 15:47 77 bytes
.vimrc 2017-12-17 15:47 4.6 KB
README.txt 2017-12-17 15:47 235 bytes
Upload a file


pythonanywhere
Dashboard Consoles Files Web Tasks Databases
/home/yiusin/ mysite
Open Bash console here
0% full – 148.0 KB of your 512.0 MB quota
Directories
Files
Enter new directory name
New directory
Enter new file name, eg hello.py
New file
static/
templates/
flask_app.py
flask_app.pyc
2018-01-29 12:45 403 bytes
2018-01-29 12:45 784 bytes
Upload a file

```
from flask import Flask, render_template

app = Flask(__name__)
@app.route("/")
def hello():
    return "Hello World!"

if __name__ == '__main__':
    app.run(debug=True)
```

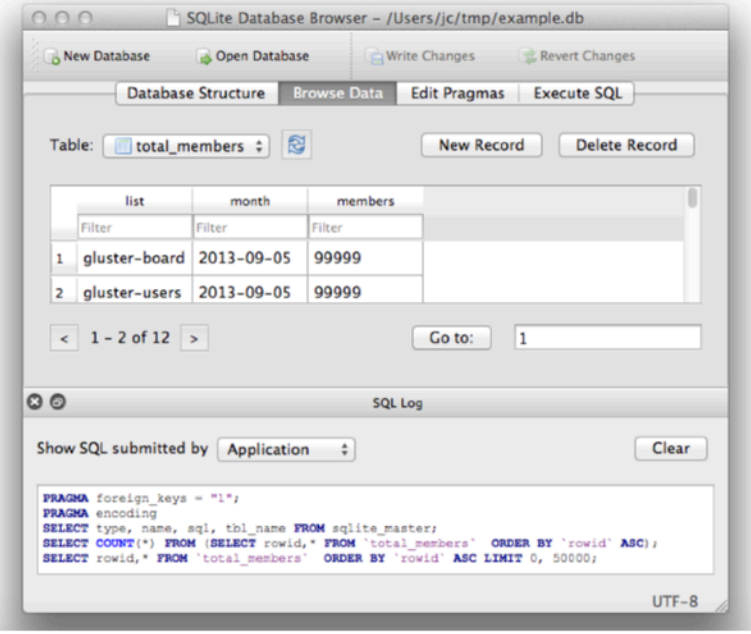
Running the application remotely at port 80 (which is the default web port that doesn't need to be shown)

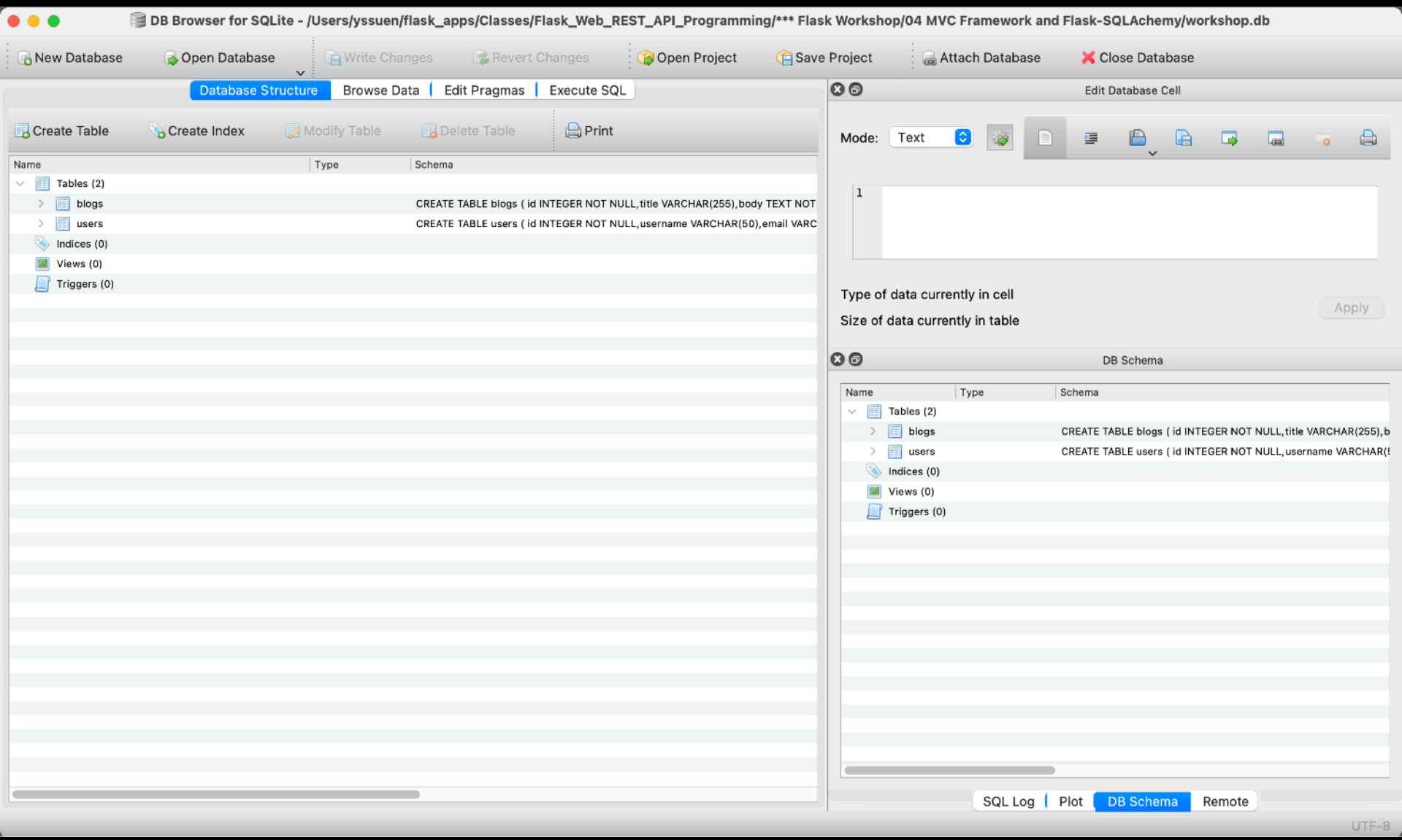
Integrating SQLite data into Flask HTML templates.

DB Browser for SQLite

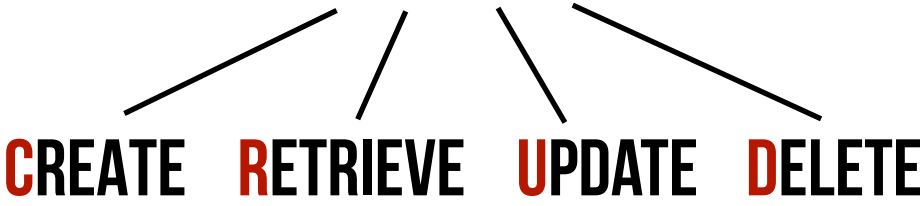
The Official home of the DB Browser for SQLite

Screenshot




DB Browser for SQLite - /Users/yssuen/flask_apps/Classes/Flask_Web_REST_API_Programming/*** Flask Workshop/04 MVC Framework and Flask-SQLAchemy/workshop.db
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Create Table
Create Index
Modify Table
Delete Table
Print
Name
Type
Schema
Tables (2)
blogs
CREATE TABLE blogs (id INTEGER NOT NULL,title VARCHAR(255),body TEXT NOT
users
CREATE TABLE users (id INTEGER NOT NULL,username VARCHAR(50),email VARC
Indices (0)
Views (0)
Triggers (0)
Edit Database Cell
Mode:
Text
1
Type of data currently in cell
Size of data currently in table
Apply
DB Schema
Name
Type
Schema
Tables (2)
blogs
CREATE TABLE blogs (id INTEGER NOT NULL,title VARCHAR(255),b
users
CREATE TABLE users (id INTEGER NOT NULL,username VARCHAR(
Indices (0)
Views (0)
Triggers (0)
SQL Log
Plot
DB Schema
Remote
UTF-8

CRUD Operations in SQL



Introduction to SQLAlchemy

Why SQLAlchemy?

- Most popular database package for Flask with extensive supports and examples.
- Support many popular SQL databases such as SQLite, MySQL, Postgresql, MS-SQL and Oracle.
- Support both Object Relational Mapping (ORM) and Structured Query Language (SQL) based data manipulation.
- Support paging of data records, making it ideal for e-commerce catalog production.
- Support one-to-many and many-to-many relationships (https://flask-sqlalchemy.palletsprojects.com/en/2.x/models/)

See Examples in Jupyter Notebook.

```python

@app.route("/blogs")
def blogs():
    df = pd.read_csv("weekly_hours.csv")
    list = df.to_dict('records')
    return render_template('blogs.html', entries = list)

#@app.route("/blogs")
#def blogs():
#    result = db.engine.execute('select * from weekly_hours')
#    weekly_hours = []
#    for i in result:          # Loop through SQL query result and add it to a users list
#        weekly_hours.append(i)
    # print(weekly_hours)
#    dataset = []                        # Define a list called dataset
#    wk_hours_rec={}               # Define a dictionary called user_rect to store individual user record
#    for i in weekly_hours:              # Loop through the weekly_hours list to assign each list element to the indiv
#        wk_hours_rec['Jan'] = i[0]        # Field by field assignment for storing list element as dictionary value
#        wk_hours_rec['Feb'] = i[1]        # Four value pairs
#        wk_hours_rec['Mar'] = i[2]
#        wk_hours_rec['Student'] = i[3]
        # print(i)                        # Print out each wk_hours_rec list element
        # print(user_rec)                 # Print out each wk_hours_rec value pair
        # dataset.append(wk_hours_rec.copy())# Append each wk_hours_rec record to the dataset list
    # print(dataset)
#    return render_template('blogs.html', entries = dataset)

```

```
{% extends "base.html" %}
{% block header %}
  <!-- Page Header-->
  <header class="masthead" style="background-image: url({{ url_for('static', filename='assets/img/home-bg.jpg') }})">
    <div class="overlay"></div>
    <div class="container">
        <div class="row">
            <div class="col-lg-8 col-md-10 mx-auto">
                <div class="site-heading">
                    <h1>Clean Blog</h1>
                    <span class="subheading">A Blog Theme by Start Bootstrap</span>
                </div>
            </div>
        </div>
    </div>
</header>
{% endblock %}
{% block content %}
<!-- Main Content-->
<div class="container">
     <div class="row">
         <div class="col">
             <b>Name</b>
        </div>
        <div class="col">
            <b>Jan</b>
        </div>
        <div class="col">
            <b>Feb</b>
         </div>
        <div class="col">
            <b>Mar</b>
        </div>
     </div>
    {% for row in  entries  %}
    <div class="row">
        <div class="col">
                  {{ row["Student"] }}
        </div>
        <div class="col">
                  {{ row["Jan"] }}
        </div>
        <div class="col">
                  {{ row["Feb"] }}
        </div>
        <div class="col">
                  {{ row["Mar"] }}
        </div>
    </div><!-- end row -->
    {% endfor %}
```

Use the for and endfor template looping command to display entries on the web page.

```
35
36 #@app.route("/blogs")
37 #def blogs():
38 #    df = pd.read_csv("weekly_hours.csv")
39 #    list = df.to_dict('records')
40 #    return render_template('blogs.html', entries = list)
41
42 @app.route("/blogs")
43 def blogs():
44     result = db.engine.execute('select * from weekly_hours')
45     weekly_hours = []
46     for i in result:          # Loop through SQL query result and add it to a users list
47         weekly_hours.append(i)
48     # print(weekly_hours)
49     dataset = []                      # Define a list called dataset
50     wk_hours_rec={}               # Define a dictionary called user_rect to store individual user record
51     for i in weekly_hours:            # Loop through the weekly_hours list to assign each list element to the indiv
52         wk_hours_rec['Jan'] = i[0]       # Field by field assignment for storing list element as dictionary value p
53         wk_hours_rec['Feb'] = i[1]       # Four value pairs
54         wk_hours_rec['Mar'] = i[2]
55         wk_hours_rec['Student'] = i[3]
56         # print(i)                       # Print out each wk_hours_rec list element
57         # print(user_rec)                # Print out each wk_hours_rec value pair
58         dataset.append(wk_hours_rec.copy())# Append each wk_hours_rec record to the dataset list
59     # print(dataset)
60     return render_template('blogs.html', entries = dataset)
61
```

Name	Jan	Feb	Mar
David Chan	90	85	88
Peter Wong	72	75	999
John Lui	60	80	100

Setup Database

In [ ]:

```
# Create database and table using SQLAlchemy and Python module
from db_schema import * # import data_model.py module with all functions included.
try:
    db.create_all()
    print("Database successfully created.")
except Exception as e:
    print("Exception occurred.",e)
```

MySQL Data Manipulation Through SQLAlchemy

In [4]:

```
from flask import Flask                          # Import the Flask module from the flask package
from flask_sqlalchemy import SQLAlchemy          # Import the SQLAlchemy module from the flask_sqlalchemy package

app = Flask(__name__)                                                  # Declare that this is a Flask app
db = SQLAlchemy(app)                                                   # Initialize the SQLAlchemy flask database obje
app.config['SQLALCHEMY_DATABASE_URI']  = 'sqlite:///crud_demo.db'      # Setup the database file "crud_demo.db"
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False                   # Disable modification track
app.config['SECRET_KEY'] = 'mysecretkey'                               # Initialize SECRET_KEY used to encrypt your co
                                                                       # and save send them to the browser for sessio

class Users(db.Model):                                                  # Define User table class and call it 'users'
    __tablename__ = 'users'                                             # Associate it with the "users" table in the d
    id = db.Column(db.Integer, primary_key=True, autoincrement=True)    # Define primary key id.
    username = db.Column(db.String(50))                                 # Define other fields.
    email = db.Column(db.String(255))
    password = db.Column(db.String(80))
```

jupyter db_schema.py 4 hours ago

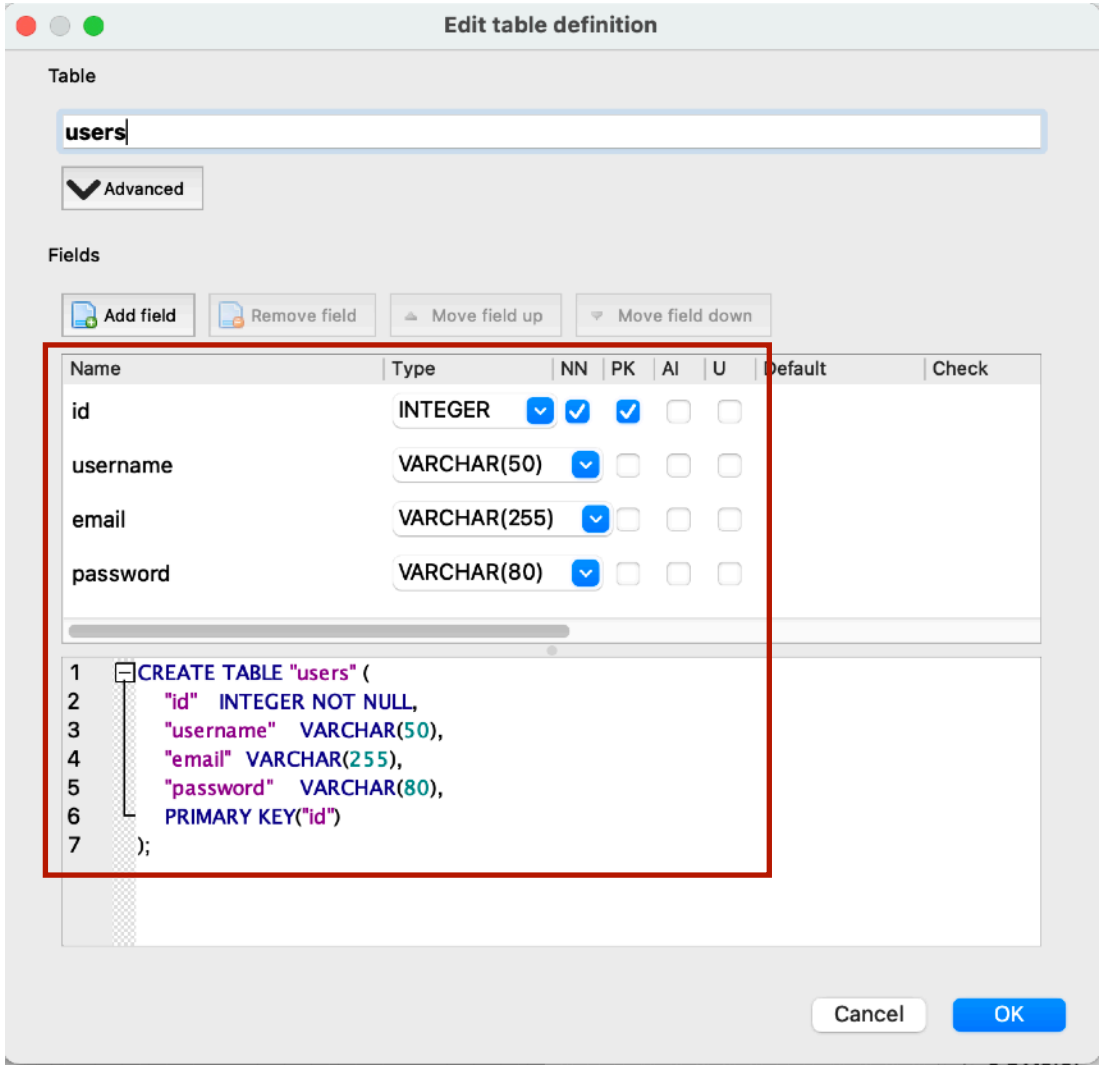
Logout

File Edit View Language Python

```python
from flask import Flask                          # Import the Flask module from the flask package
from flask_sqlalchemy import SQLAlchemy          # Import the SQLAlchemy module from the flask_sqlalchemy package

app = Flask(__name__)                                                  # Declare that this is a Flask app
db = SQLAlchemy(app)                                                   # Initialize the SQLAlchemy flask database object
app.config['SQLALCHEMY_DATABASE_URI']  = 'sqlite:///crud_demo.db'      # Setup the database file "crud_demo.db"
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False                   # Disable modification track
app.config['SECRET_KEY'] = 'mysecretkey'                               # Initialize SECRET_KEY used to encrypt your cookies
                                                                       # and save send them to the browser for session management.

class User(db.Model):                                                  # Define User table class and call it 'users'
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True) # Define primary key id.
    username = db.Column(db.String(50))                                 # Define other fields.
    email = db.Column(db.String(255))
    password = db.Column(db.String(80))
```


Edit table definition
Table
users
Advanced
Fields
Add field
Remove field
Move field up
Move field down
Name
Type
NN
PK
AI
U
Default
Check
id
INTEGER
username
VARCHAR(50)
email
VARCHAR(255)
password
VARCHAR(80)
1 CREATE TABLE "users" (
2 "id" INTEGER NOT NULL,
3 "username" VARCHAR(50),
4 "email" VARCHAR(255),
5 "password" VARCHAR(80),
6 PRIMARY KEY("id")
7);
Cancel
OK

Create MySQL Record Through SQLAlchemy Data Object

In [ ]:
```
# Create entry
# product.productID, product.productCode, product.name, product.quantity, product.price,product.supplierID
user = Users(id=1003,username="Kaki",email="kaki@cuhk.edu.hk",password='12345')
# Add user and commit entry
db.session.add(user)
db.session.commit()
```

Retrieve MySQL Record Through SQLAlchemy Data Object and SQL Query

In [6]:
```
users = Users.query.order_by(Users.id.desc()).all() # Setup users query (equivalent to SELECT * FROM Users
                                                    # ORDER BY id DESC)
for user in users:                                  # Loop through all users entry by entry
    print(user.id, user.username, user.email)       # Print each one out
```

```
1002 John johnsmith@cuhk.edu.hk
1001 Meimei meimei@cuhk.edu.hk
```

In [9]:
```
result = db.engine.execute('select * from users')
for i in result:
    print(i)
```

```
(1001, 'Meimei', 'meimei@cuhk.edu.hk', 'mypass')
(1002, 'John', 'johnsmith@cuhk.edu.hk', 'mypass')
```

In [8]:
```
result = db.engine.execute('select * from users')
users = []
for i in result:          # Loop through SQL query result and add it to a users list
    users.append(i)
# print(users)
dataset = []              # Define a list called dataset
user_rec={}               # Define a dictionary called user_rect to store individual user record
for i in users:           # Loop through the users list to assign each list element to the individual user record
    user_rec['id'] = i[0]          # Field by field assignment for storing list element as dictionary value pair
    user_rec['username'] = i[1]    # Three value pairs id/id value, username/username value, email/email value
    user_rec['email'] = i[2]
    # print(i)                       # Print out each users list element
    # print(user_rec)                # Print out each user record value pair
    dataset.append(user_rec.copy())# Append each user record to the dataset list
print(dataset)
```

Week_Hours data migration from CSV file to SQL table

In [10]:

```
import pandas as pd
df = pd.read_csv("weekly_hours.csv")
list = df.to_dict('records')
print(list)
```

```
[{'Jan': 85, 'Feb': 90, 'Mar': 88, 'Student': 'David Chan'}, {'Jan': 75, 'Feb': 72, 'Mar': 999, 'Student': 'Peter Won
g'}, {'Jan': 80, 'Feb': 60, 'Mar': 100, 'Student': 'John Lui'}]
```

In [12]:

```
import pandas as pd                                    # Import pandas library package
import sqlite3                                         # Import SQLite3 library package
con = sqlite3.connect('crud_demo.db')                  # Connect to SQLite database using sqlite3 connect method
# load the data into a Pandas DataFrame.
weekly_hours = pd.read_csv('weekly_hours.csv')         # Read csv into a panda dataframe called weekly_hours
print(weekly_hours) # print out weekly_hours dataframe
cursor = con.cursor()                                  # Setup record pointer cursor to prepare execution
drop_sql = "DROP TABLE IF EXISTS weekly_hours"         # Drop weekly_hours table to prepare for new import
try:
    cursor.execute(drop_sql)
    con.commit()
    print("Table successfully dropped.")
except Exception as e:
    print("Exception occurred.",e)

# write the data to a sqlite table.
weekly_hours.to_sql('weekly_hours', con, if_exists='append', index = False) # Save weekly_hours
                                                                            # dataframe to SQL table
sql = "SELECT * FROM weekly_hours"                     # Set up SQL query statement
df = pd.read_sql_query(sql,con)                        # Read result into dataframe
con.close()                                            # Close connection
df                                                     # Display data frame
```

```
   Jan  Feb  Mar      Student
0   85   90   88  David Chan
1   75   72  999  Peter Wong
2   80   60  100    John Lui
Table successfully dropped.
```

Out[12]:

	Jan	Feb	Mar	Student
0	85	90	88	David Chan
1	75	72	999	Peter Wong
2	80	60	100	John Lui

In [14]:
```
result = db.engine.execute('select * from weekly_hours')
weekly_hours = []
for i in result:          # Loop through SQL query result and add it to a users list
    weekly_hours.append(i)
# print(weekly_hours)
dataset = []                        # Define a list called dataset
wk_hours_rec={}                # Define a dictionary called user_rect to store individual user record
for i in weekly_hours:              # Loop through the weekly_hours list to assign each list element to the individua
    wk_hours_rec['Jan'] = i[0]        # Field by field assignment for storing list element as dictionary value pair
    wk_hours_rec['Feb'] = i[1]        # Four value pairs
    wk_hours_rec['Mar'] = i[2]
    wk_hours_rec['Student'] = i[3]
    # print(i)                        # Print out each wk_hours_rec list element
    # print(user_rec)                 # Print out each wk_hours_rec value pair
    dataset.append(wk_hours_rec.copy())# Append each wk_hours_rec record to the dataset list
print(dataset)
```

```
[{'Jan': 85, 'Feb': 90, 'Mar': 88, 'Student': 'David Chan'}, {'Jan': 75, 'Feb': 72, 'Mar': 999, 'Student': 'Peter Won
g'}, {'Jan': 80, 'Feb': 60, 'Mar': 100, 'Student': 'John Lui'}]
```

Update MySQL Record Through SQLAlchemy Data Object

In [ ]:
```
# Update entry
user = Users.query.filter_by(id=1003).first()
user.username = 'Kaki_Li'
user.email = 'kaki_li@cuhk.edu.hk'
db.session.commit()
```

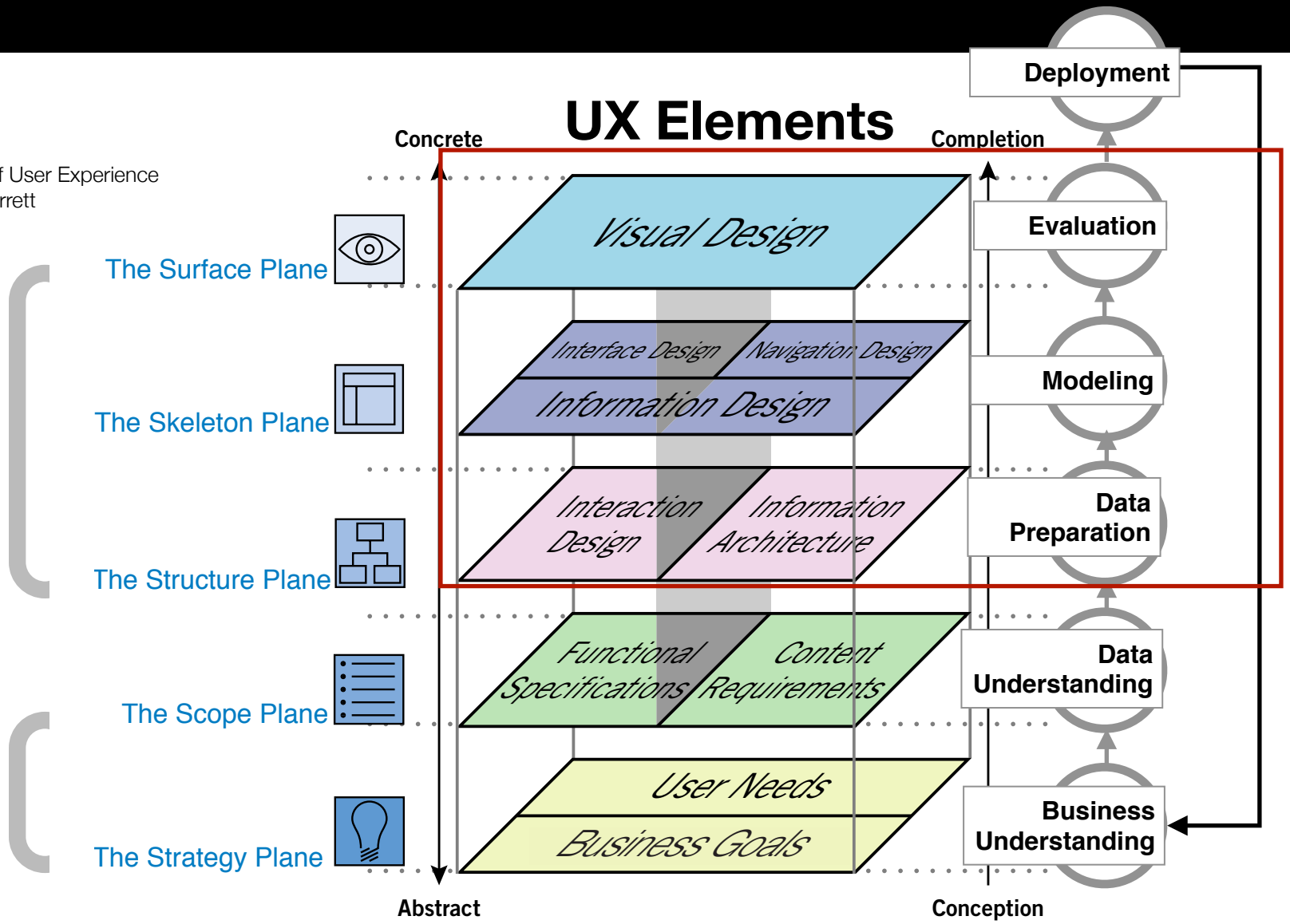
Delete MySQL Record Through SQLAlchemy Data Object

In [ ]:
```
# Delete entry
user = Users.query.filter_by(id=1003).first()
db.session.delete(user)
db.session.commit()
```

Usability tests for future UX hypothesis validation using Figma wire frames.

UX Elements

Source: Elements of User Experience
by Jesse James Garrett



Business Goals

UX AS A JOURNEY

Elements

1. Persona
2. Context
3. Artifacts
4. Tasks/Jobs

Structure

1. Problem
2. Alternatives
3. Transformation
4. Synthesis

} Current Journey

} Future Journey

1. **Problem**
2. **Alternatives**

} Current Journey

3. **Transformation**
4. **Synthesis**

} Future Journey

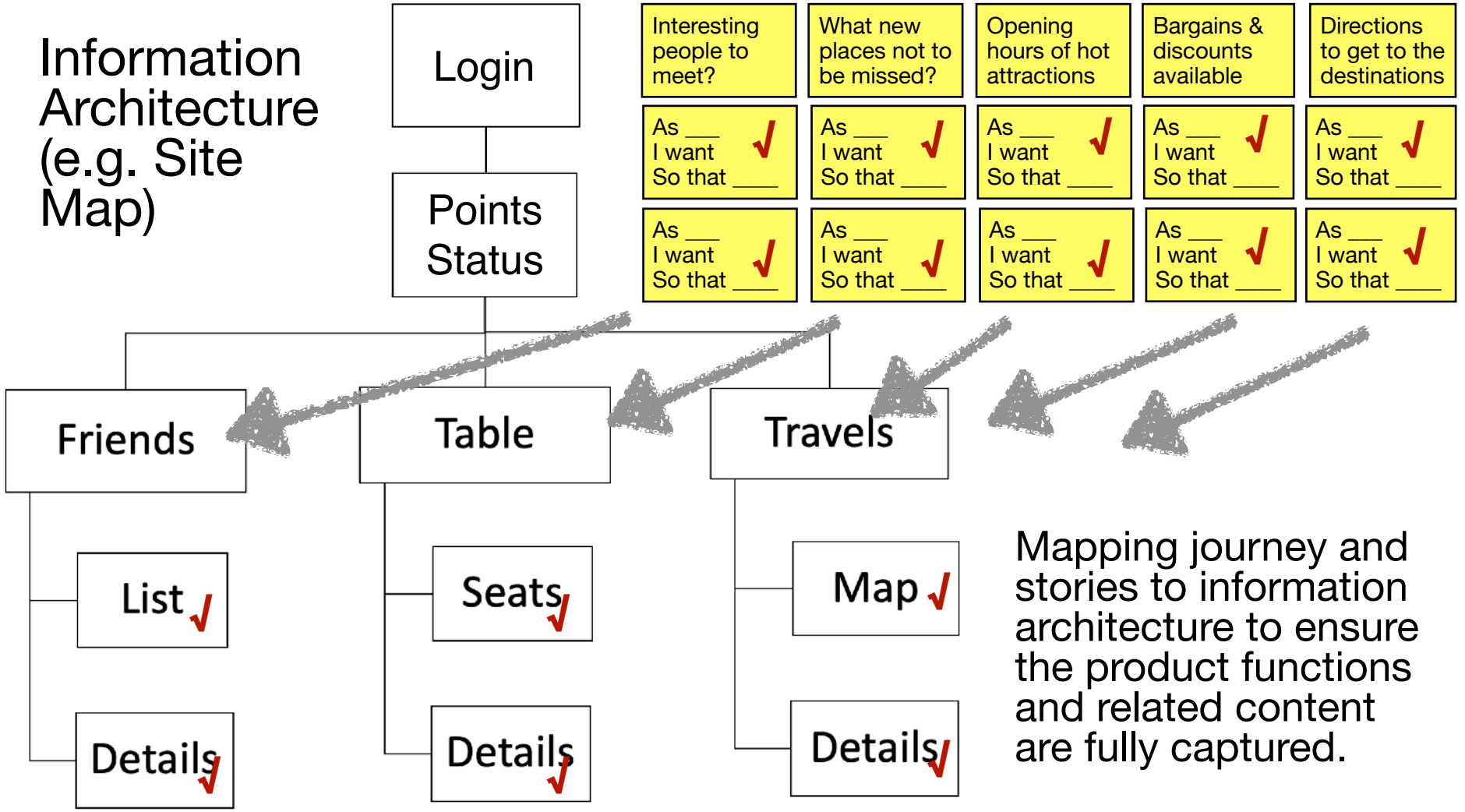
"Problem" Hypothesis to be validated.

- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓

"Solution" Hypothesis to be validated.

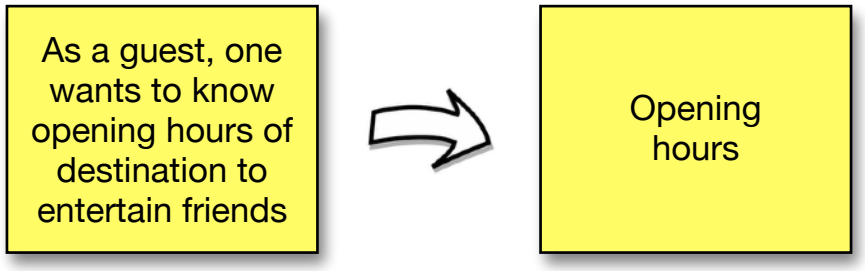
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓

Information Architecture (e.g. Site Map)

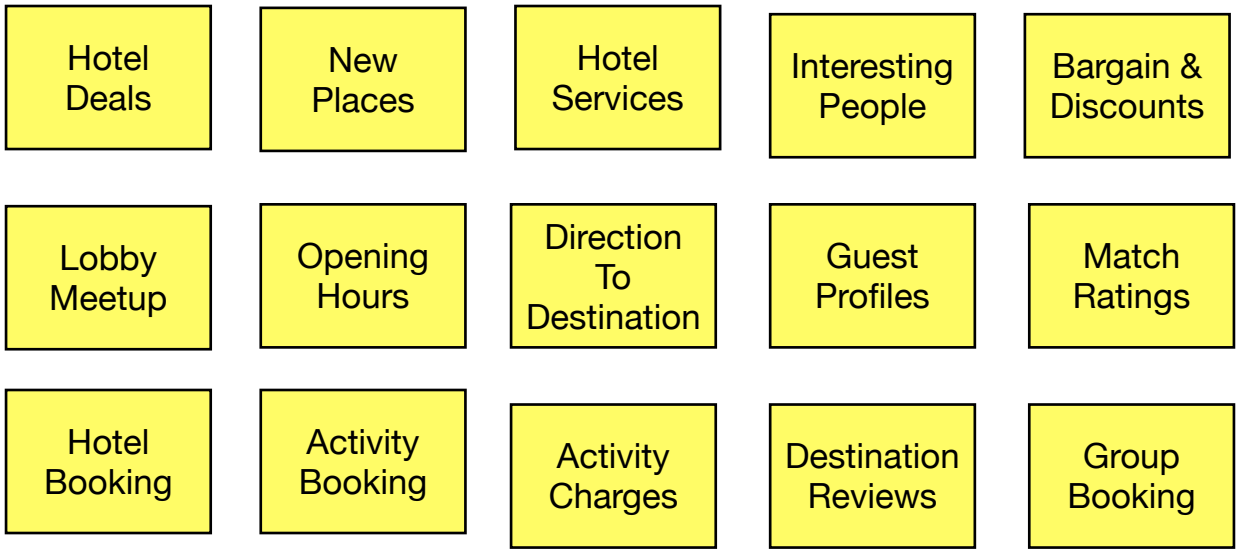


Mapping journey and stories to information architecture to ensure the product functions and related content are fully captured.

From Phase to Noun



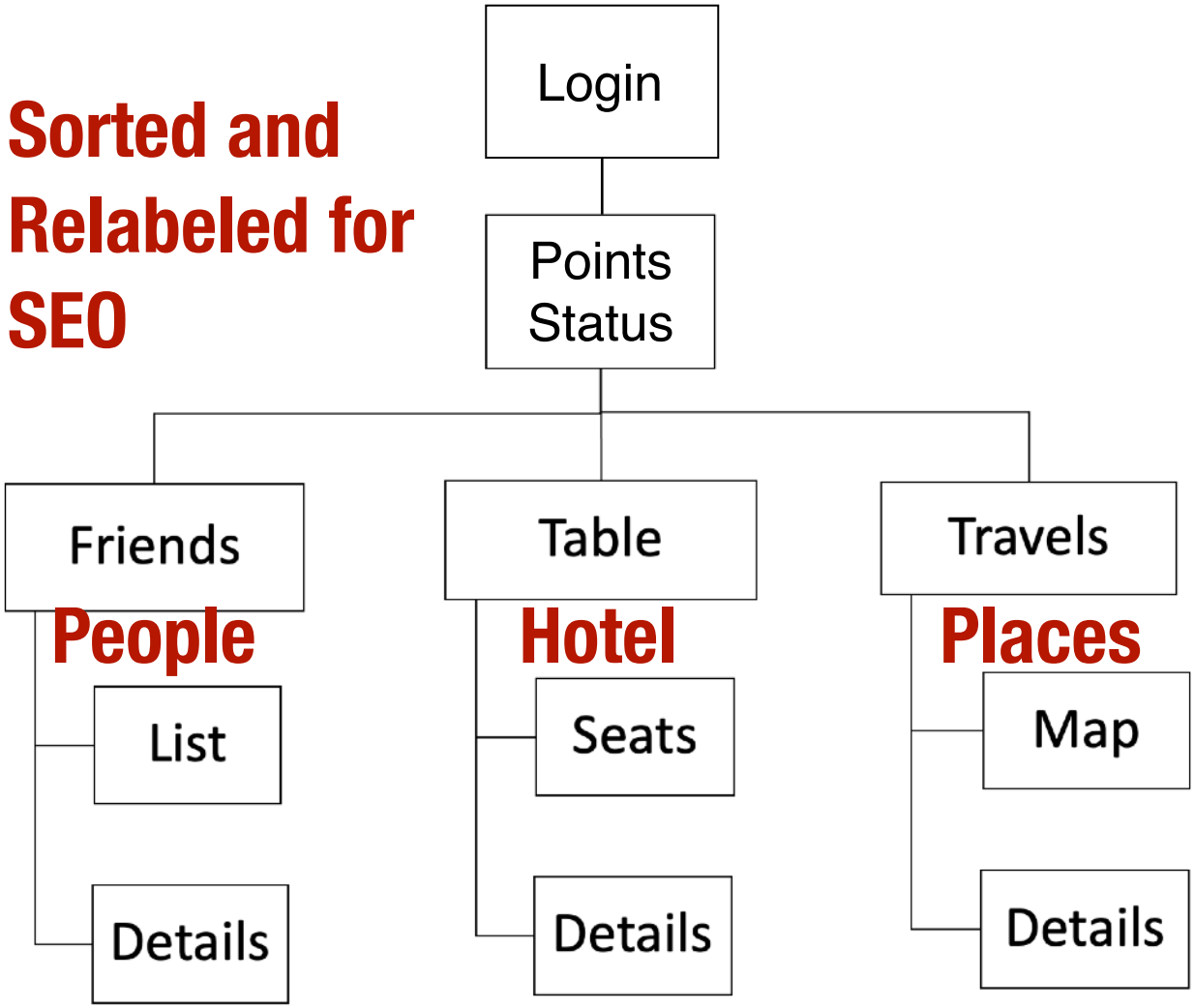
Card Sorting (Before)

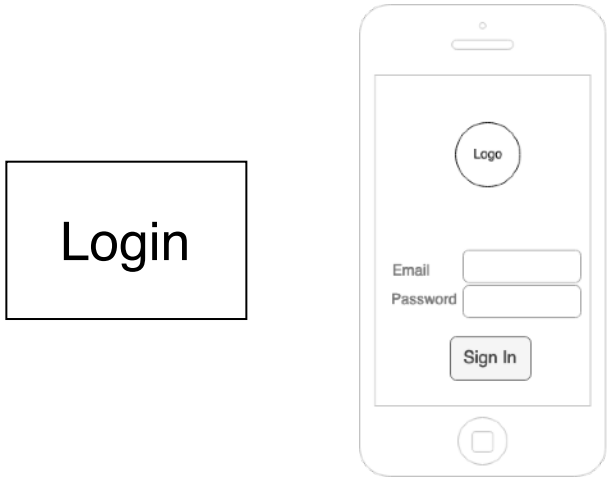


Card Sorting (After)

Hotel		Places		People
Hotel Services	Hotel Deals	Opening Hours	Bargain & Discounts	Interesting People
Hotel Booking	Lobby Meetup	New Places	Direction To Destination	Guest Profiles
Group Booking	Activity Booking	Destination Reviews	Activity Charges	Match Ratings

Sorted and Relabeled for SEO



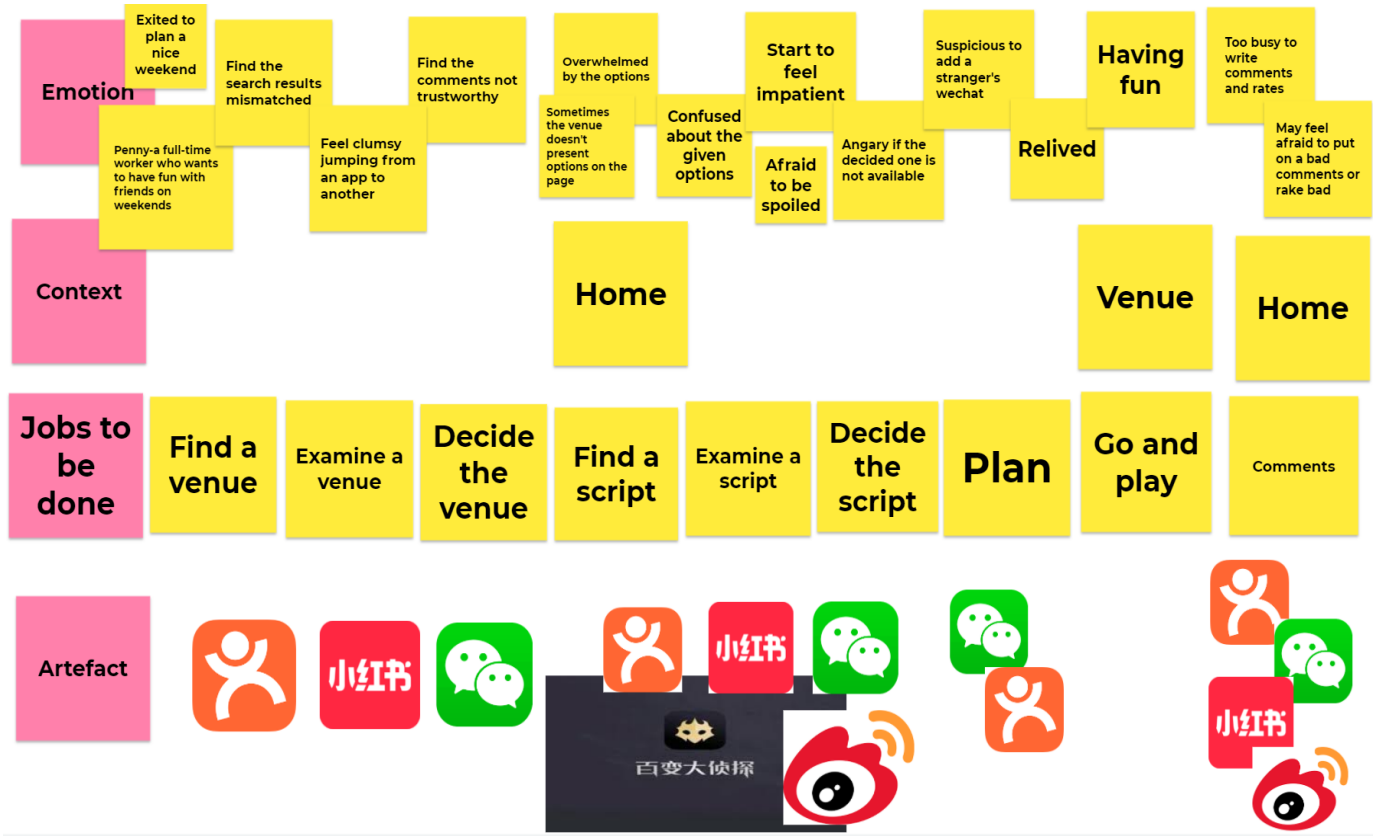


Wireframe

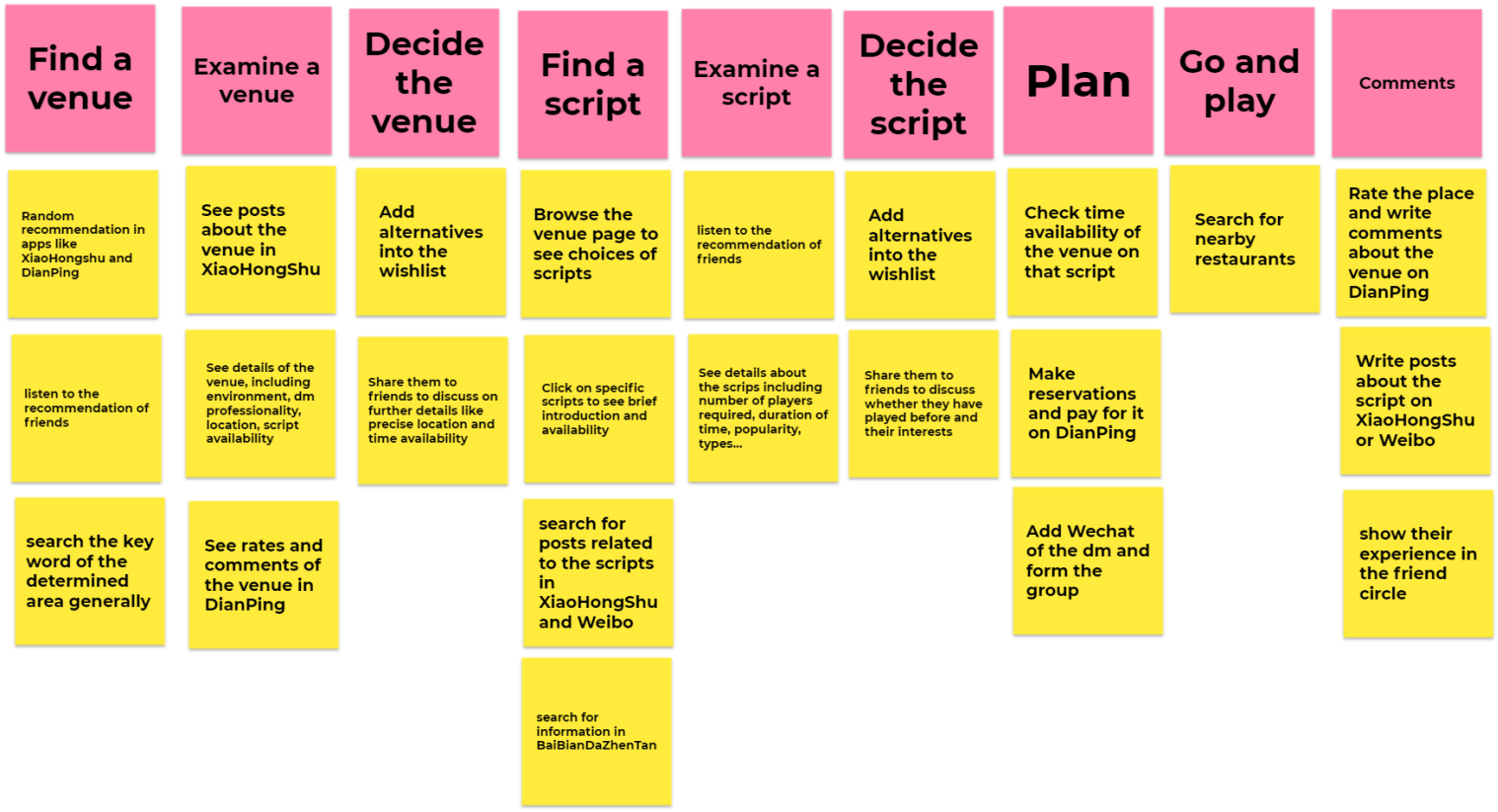
Example


主页 剧本 场馆 约本 注册 登录 关于
温州剧本杀俱乐部
这里有所有你想要的信息，一起来约本吧！

Current Customer Journey Map



Current Story Map

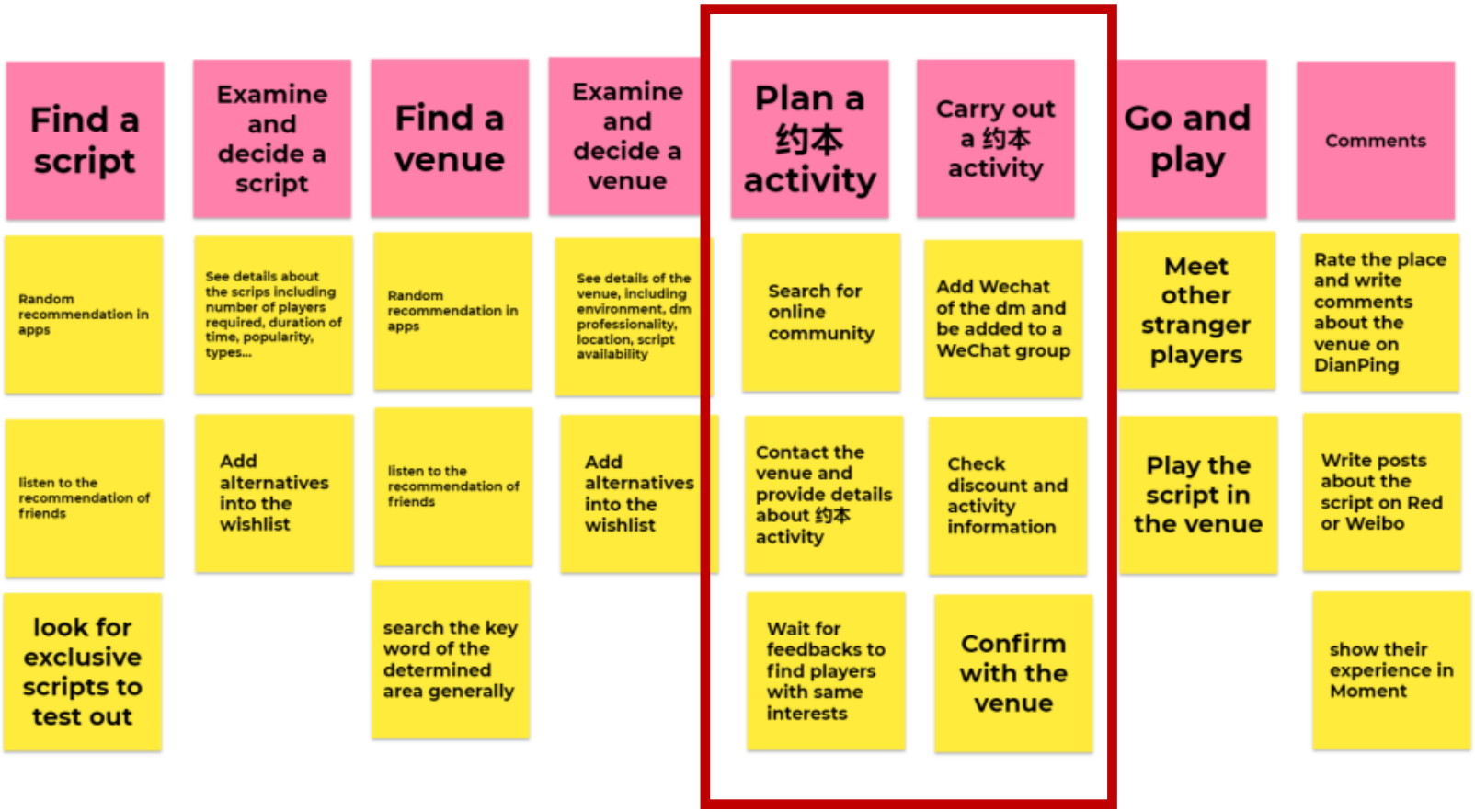


Functional	**Overall Venue Information**	**Detailed Venue Information**	**Script Information**	**Planification and Confirmation**	**Comments**
Content	*Location:* 1.Is it within the area required? 2.Is it easy for everyone to go? How far is the distance?	*Location:* 1.Detailed location 2.Possible ways of getting there	*Attributes:* 1.Scripts type 2.Number of players required 3.Scripts introduction 4.Duration of time	*Economics:* 1.Cost 2.Payment methods	1.Which platform to put on comments? 2.Is it safe to write bad comments?
	Service: 1.Does it have lots of scripts? 2. Does it provide many time options?	*Service:* 1.Details of scripts provided 2.Details of time availability	*References:* 1.Players' comments on the scripts 2.Does it up-to-date or popular?	*Others:* 1.Availability 2.Contact with the venue	
	Others： 1.Is the environment good for the gathering? 2.Is it popular on social media?	*Others:* 1.Comments about the chosen venue 2. Professionality of dm to lead the game			

Problem Statement

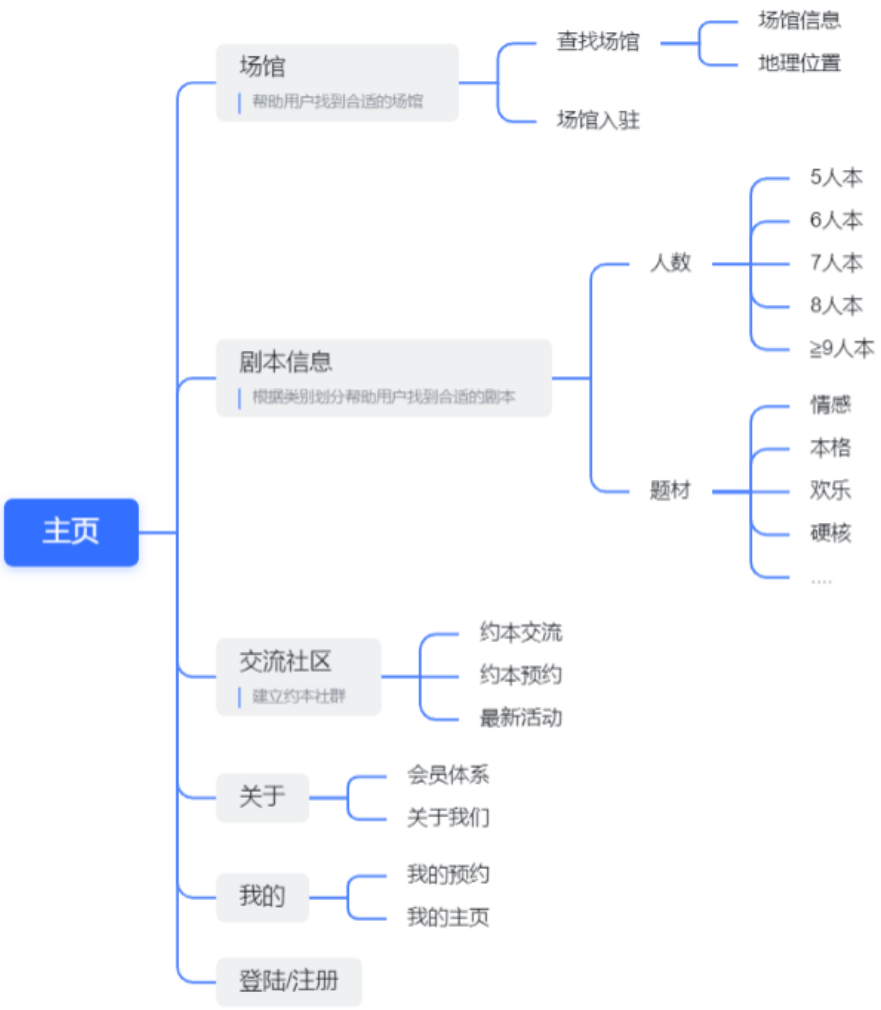
How might we make a full-time worker who wants to plan a murder mystery game with friends on weekend (who) address the problem of collecting information, generally considered as incomplete and ingenuine, on different platforms (what) so that she/he can save time and have a better murder mystery game experience (why)?

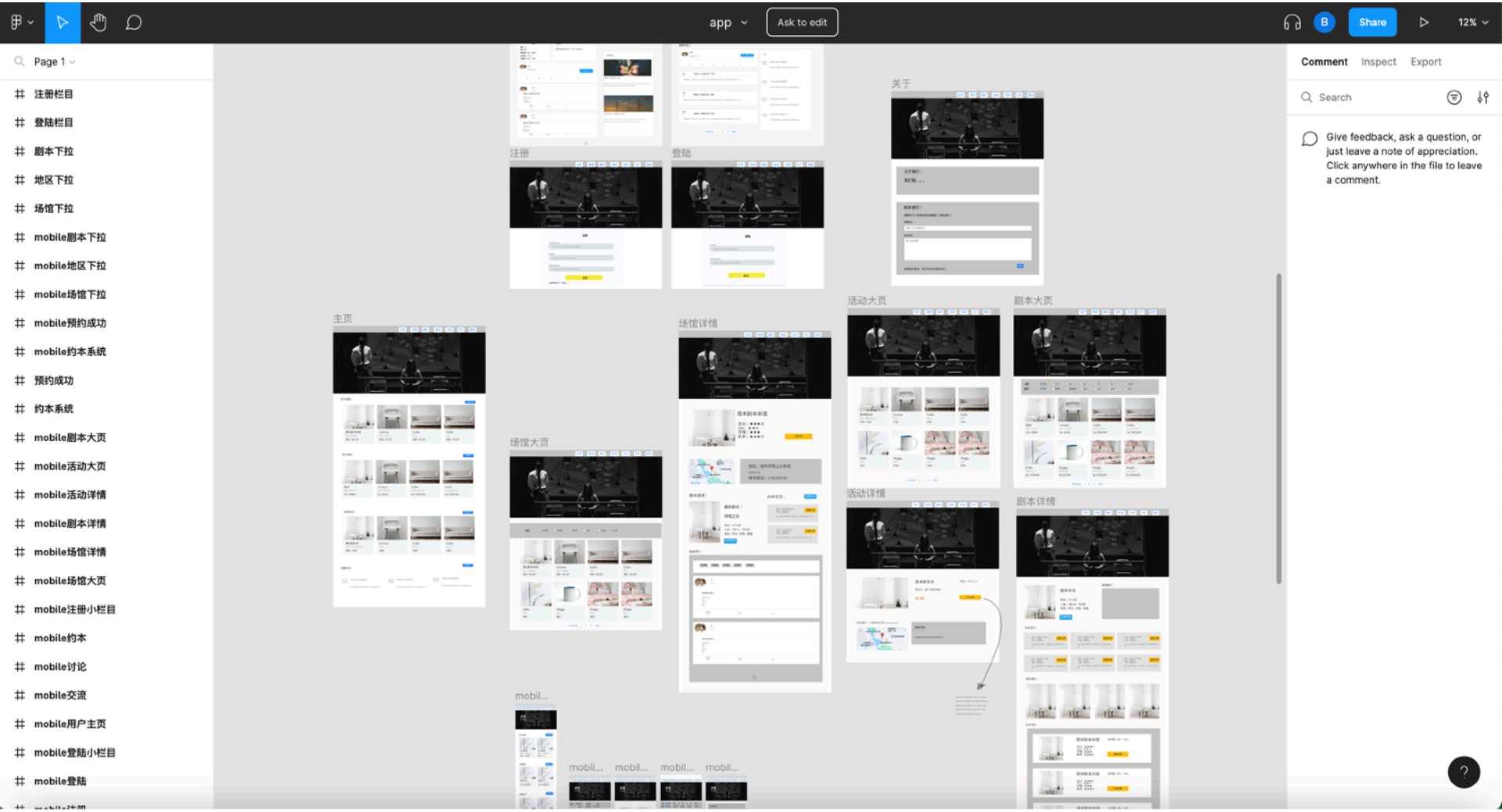
Revised Customer Journey Map & Story Map:


Find a script
Examine and decide a script
Find a venue
Examine and decide a venue
Plan a 约本 activity
Carry out a 约本 activity
Go and play
Comments
Random recommendation in apps
See details about the scrips including number of players required, duration of time, popularity, types...
Random recommendation in apps
See details of the venue, including environment, dm professionality, location, script availability
Search for online community
Add Wechat of the dm and be added to a WeChat group
Meet other stranger players
Rate the place and write comments about the venue on DianPing
listen to the recommendation of friends
Add alternatives into the wishlist
listen to the recommendation of friends
Add alternatives into the wishlist
Contact the venue and provide details about 约本 activity
Check discount and activity information
Play the script in the venue
Write posts about the script on Red or Weibo
look for exclusive scripts to test out
search the key word of the determined area generally
Wait for feedbacks to find players with same interests
Confirm with the venue
show their experience in Moment

Card sorting

Participants' test:

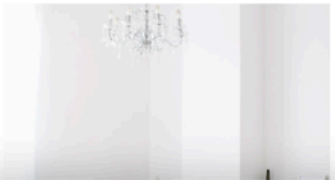

主页
场馆
帮助用户找到合适的场馆
查找场馆
场馆信息
地理位置
场馆入驻
剧本信息
根据类别划分帮助用户找到合适的剧本
人数
5人本
6人本
7人本
8人本
≥9人本
题材
情感
本格
欢乐
硬核
....
交流社区
建立约本社群
约本交流
约本预约
最新活动
关于
会员体系
关于我们
我的
我的预约
我的主页
登陆/注册


app
Ask to edit
Share
12%
Page 1
注册栏目
登陆栏目
剧本下拉
地区下拉
场馆下拉
mobile剧本下拉
mobile地区下拉
mobile场馆下拉
mobile预约成功
mobile约本系统
预约成功
约本系统
mobile剧本大页
mobile活动大页
mobile活动详情
mobile剧本详情
mobile场馆详情
mobile场馆大页
mobile注册小栏目
mobile约本
mobile讨论
mobile交流
mobile用户主页
mobile登陆小栏目
mobile登陆
注册
登陆
关于
主页
场馆大页
场馆详情
活动大页
剧本大页
活动详情
剧本详情
Comment
Inspect
Export
Search
Give feedback, ask a question, or just leave a note of appreciation. Click anywhere in the file to leave a comment.

主页 场馆 剧本 活动 交流 关于 我的

热门场馆：

查看更多



1. **Problem**
2. **Alternatives**

} Current Journey

3. **Transformation**
4. **Synthesis**

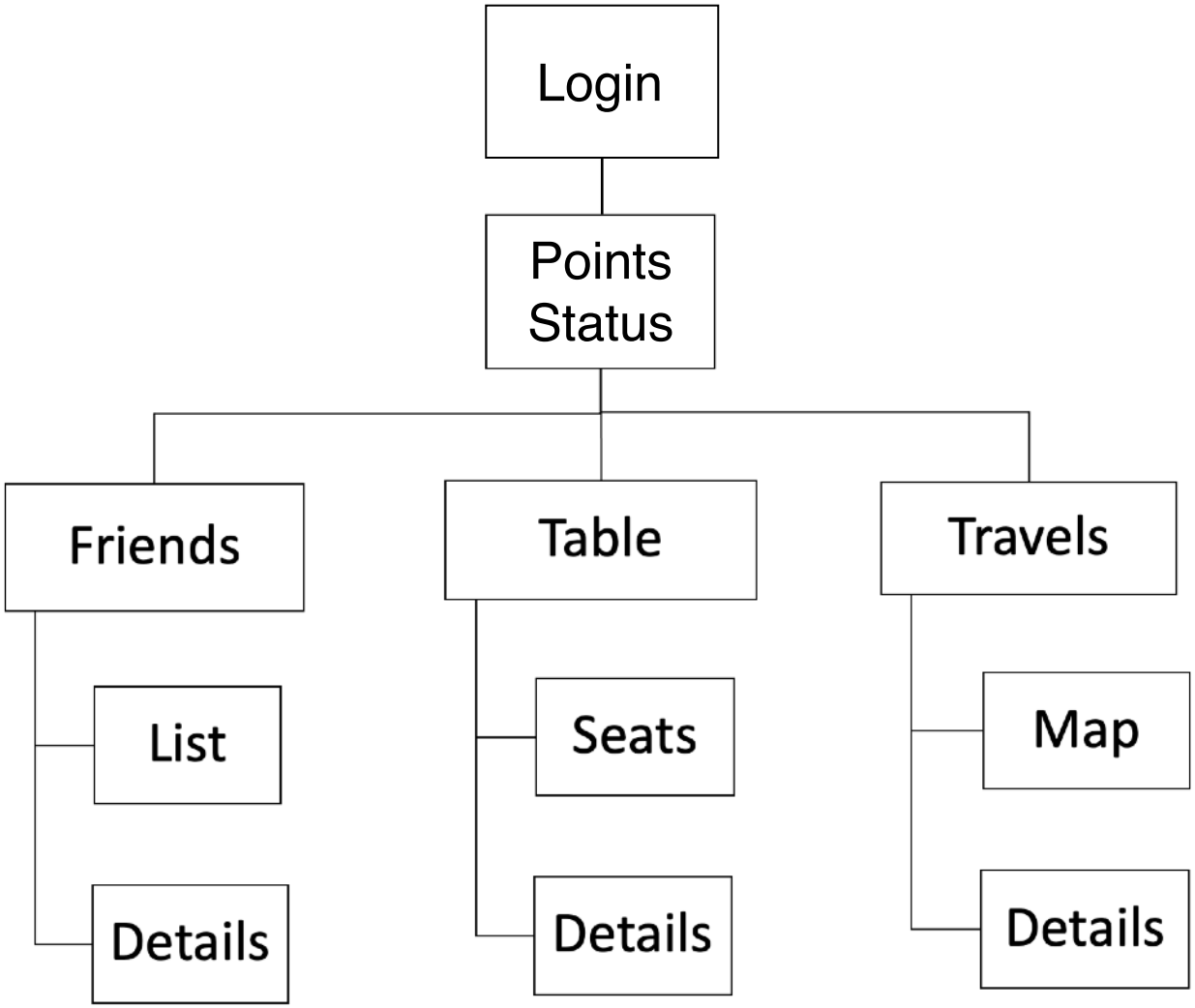
} Future Journey

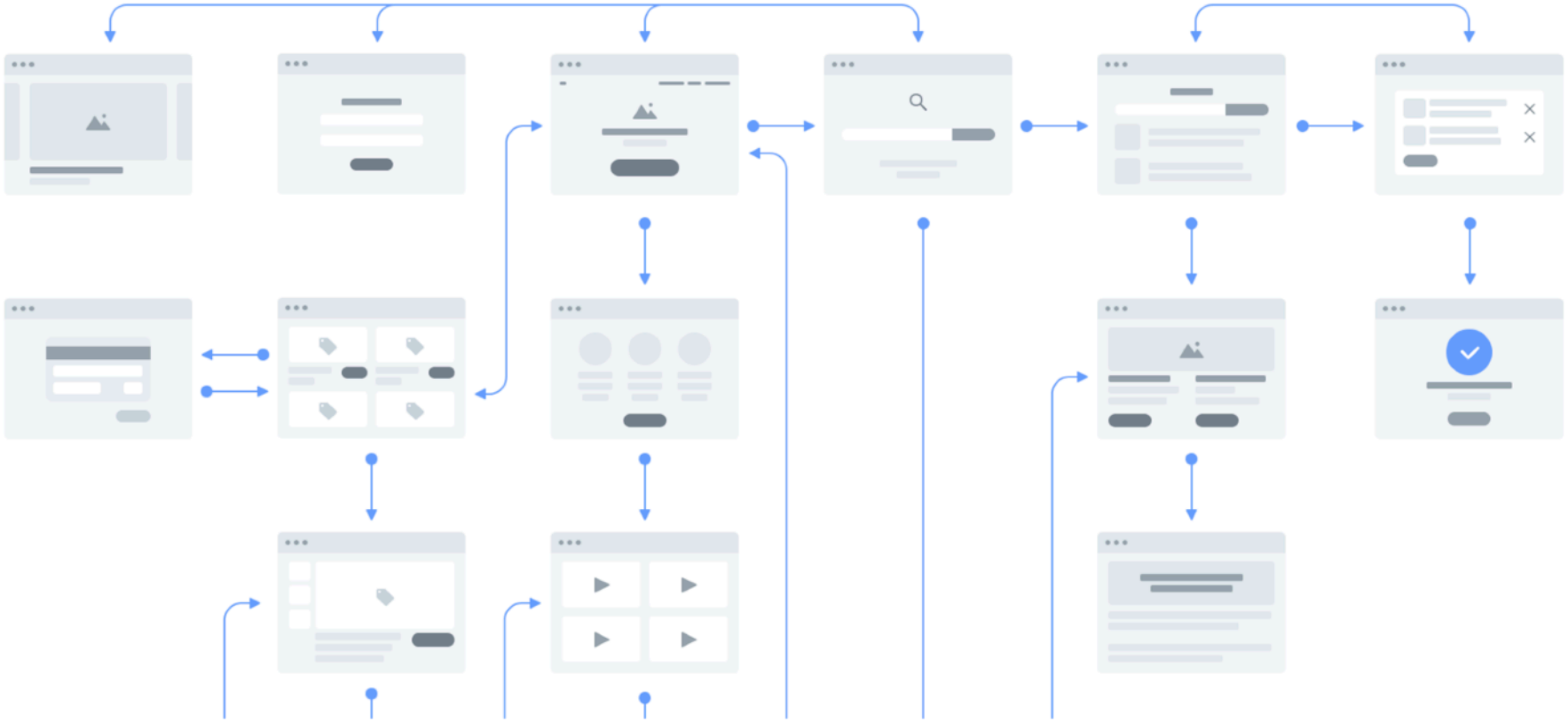
"Problem" Hypothesis to be validated.

- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓

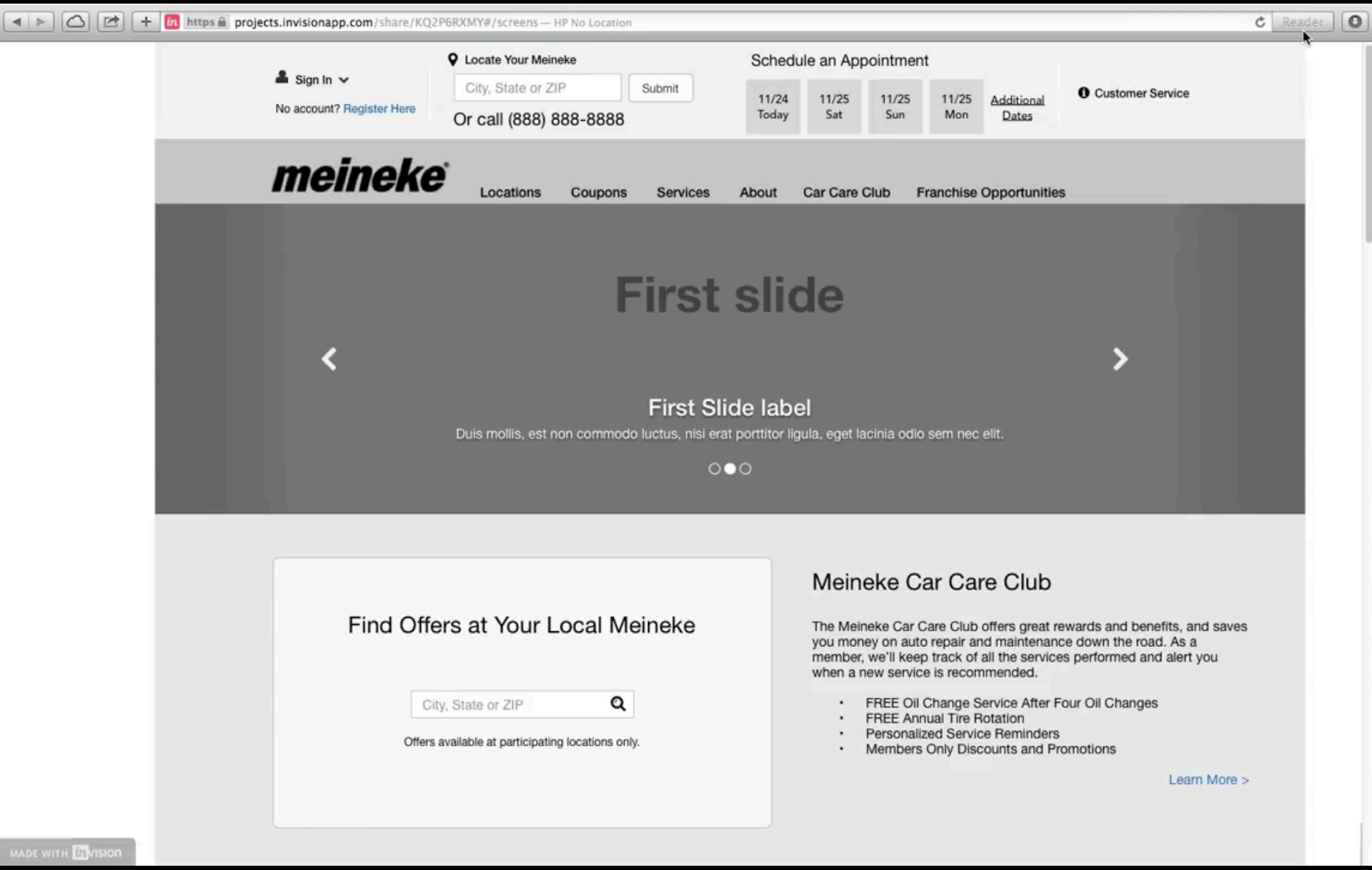
"Solution" Hypothesis to be validated.

- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓

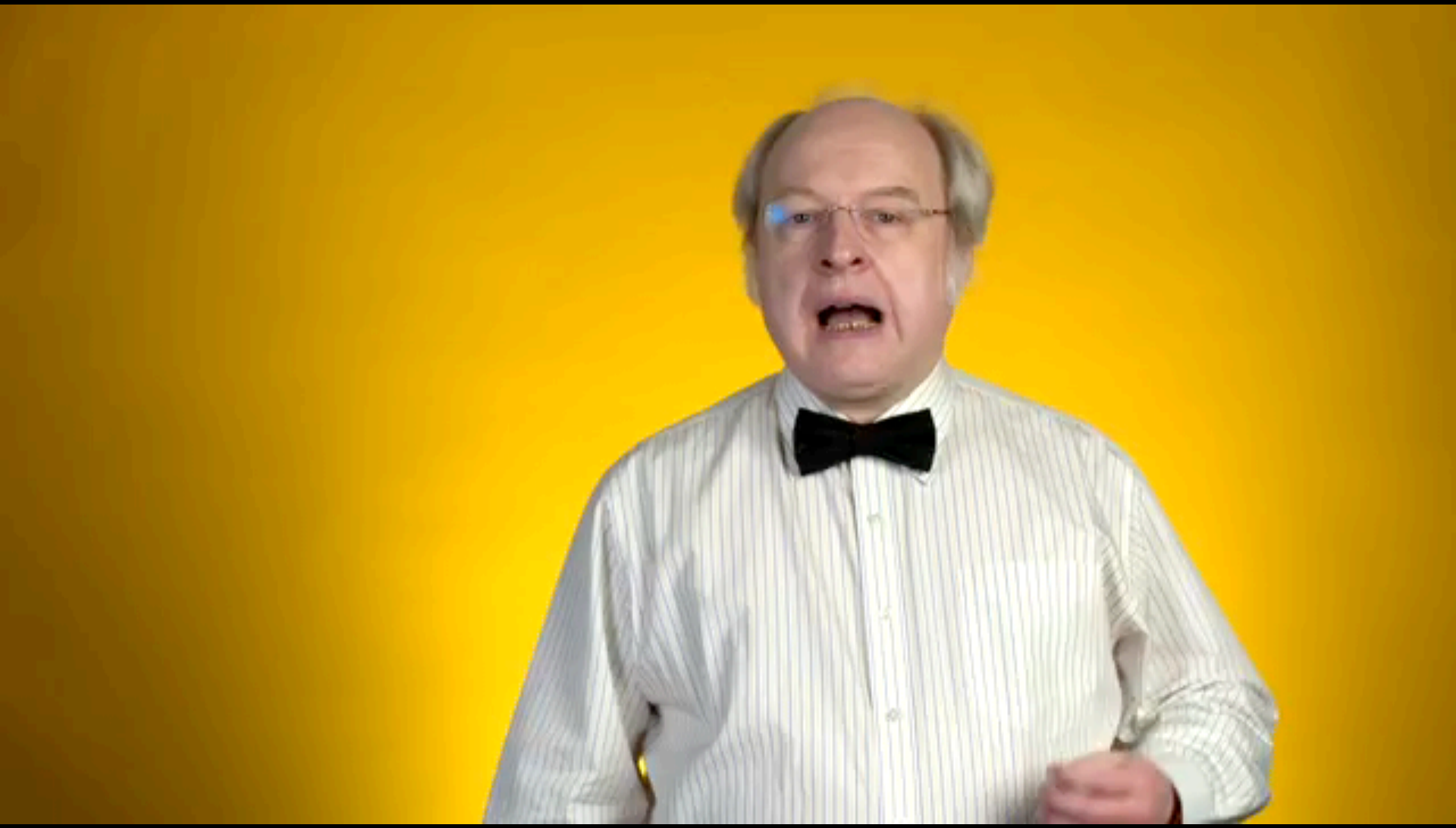

Login
Points Status
Friends
Table
Travels
List
Details
Seats
Details
Map
Details



Source: https://www.smashingmagazine.com/2020/04/wireframe-design-suc



https://www.youtube.com/watch?time_continue=1&v=thNZIZmMDQo



https://www.youtube.com/watch?v=v8JJrDvQDF4&feature=emb_logo

How to Test with Users in a Usability Study?

1. Find representative users (persona).
2. Ask to perform realistic tasks.
3. Shut up while the user is performing the task and observe.
4. Never provide any hints or leads to guide their actions. You want to observe their natural tendency.
5. Request users to speak aloud while performing the tasks. Obtain approval for recording the session.
6. Record the session (e.g. screen recording using Zoom) for later analysis.

Source: https://www.userfocus.co.uk/articles/usability_test_plan_dashboard.html

USABILITY TEST PLAN DASHBOARD

AUTHOR

CONTACT DETAILS

FINAL DATE FOR COMMENTS

PRODUCT UNDER TEST

What's being tested? What are the business and experience goals of the product?

Website [温州剧本杀俱乐部]
To provide an online community for scripts murder game fans.

BUSINESS CASE

Why are we doing this test? What are the benefits? What are the risks of not testing?

To provide insights for pm of user-friendliness and of business model as well as improve the design. If not, the product is valueless.

TEST OBJECTIVES

What are the goals of the usability test? What specific questions will be answered? What hypotheses will be tested?

Do people understand the value of the onine community?

Do people find it helpful?

Do people trust the service?

Do people understand how to make an appiontment on the platform?

PARTICIPANTS

How many participants will be recruited? What are their key characteristics?

Five.
-All are scripts murder game fans who look for community and want to 约本

EQUIPMENT

What equipment is required? How will you record the data?

Laptops with ZOOM or Tencent Meeting.

Recorder.

Timer.

TEST TASKS

What are the test tasks?

Suppose you want to make an appointment to play the game with other fans:
1. Where would you click to get information
2. How would you get more details and what details would you get
3. Find someone who can play the game with you and consider whether or not to realize a " 约本" activity
4. Register and log in
5. Fill in the appointment information and submit

RESPONSIBILITIES

Who is involved in the test and what are their responsibilities?

Tingyu ZHENG
(interviewer, recorder, task designer, all)

LOCATION & DATES

Where and when will the test take place? When and how will the results be shared?

2021/11/5-7 via ZOOM

2021/11/8 via analysis

PROCEDURE

What are the main steps in the test procedure?

0-1min: Welcome session → 1-2min: Introduction of the web → 2-3min: lend persona to introduce → 4-10: Carry out the test tasks → 10-13min: Post-test interviews → 13-15min: Express gratitute to interviewees

https://hkdesign.org/wp-content/uploads/com5961/usability_test_example.pdf

What did you notice in the usability study?

How many users should you test?

Topics

E-commerce
Intranets
Mobile & Tablet
User Testing
Web Usability
Writing for the Web
See all topics

Recent Articles

Five Mistakes in Designing Mobile Push Notifications
Filling the Silence with Digital Noise
Design Guidelines for Input Steppers
UX Debt: How to Identify, Prioritize, and Resolve
UX Guidelines for Recommended Content
See all articles

Popular Articles

10 Usability Heuristics for User Interface Design
When to Use Which User-Experience Research Methods
Usability 101: Introduction to Usability
Flat UI Elements Attract Less Attention and Cause Uncertainty
F-Shaped Pattern For Reading Web Content (original study)
Design Thinking 101
10 Best Intranets of 2017
The Distribution of Users' Computer Skills: Worse Than You Think
UX Research Cheat Sheet
When and How to Create Customer Journey Maps

Why You Only Need to Test with 5 Users

by Jakob Nielsen on March 19, 2000
Topics: User Testing

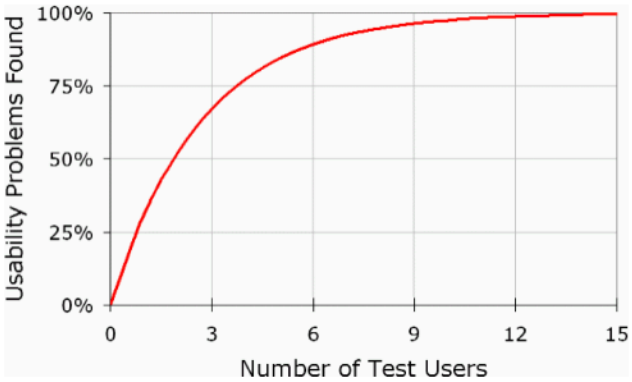
Summary: Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford.

Some people think that usability is very costly and complex and that user tests should be reserved for the rare web design project with a huge budget and a lavish time schedule. Not true. Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford.

In earlier research, Tom Landauer and I showed that the number of usability problems found in a usability test with *n* users is:

$$N(1-(1-L)^n)$$

where *N* is the total number of usability problems in the design and *L* is the proportion of usability problems discovered while testing a single user. The typical value of *L* is 31%, averaged across a large number of projects we studied. Plotting the curve for *L* =31% gives the following result:



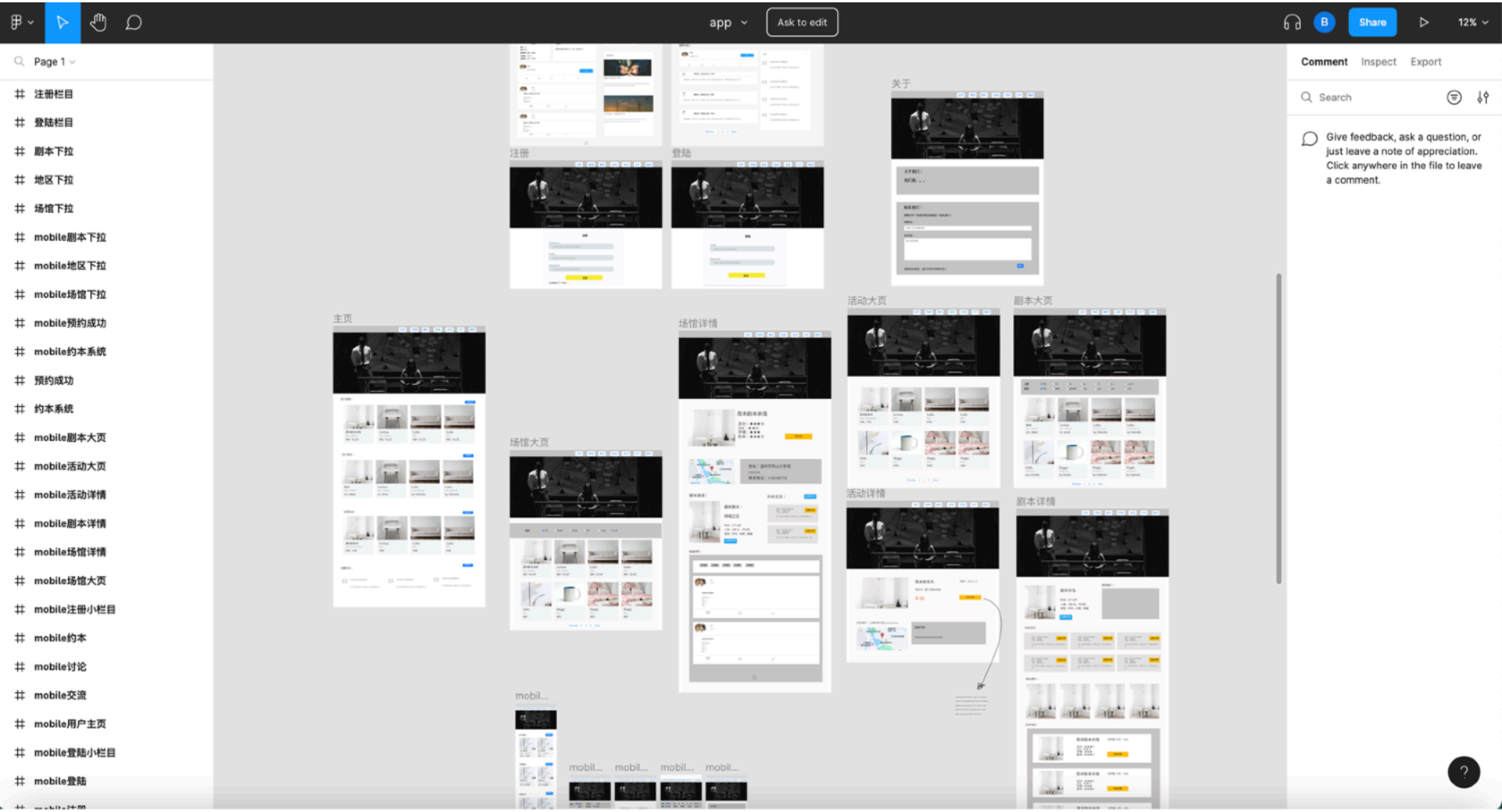
https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

https://www.youtube.com/watch?v=RhgUirqki50

Useful References

1. Usability 101 - Introduction to Usability (https://www.nngroup.com/articles/usability-101-introduction-to-usability/)
2. What is usability? (https://www.userfocus.co.uk/consultancy/usabilitytesting.html)
3. The 1 page usability test plan (https://www.userfocus.co.uk/articles/usability_test_plan_dashboard.html).
4. Usability test plan dashboard (https://www.userfocus.co.uk/pdf/usabilitydashboard.pdf)
5. Creating the usability dashboard (https://www.userfocus.co.uk/articles/dashboard.html)
6. Measure usability satisfaction (https://www.userfocus.co.uk/articles/satisfaction.html)

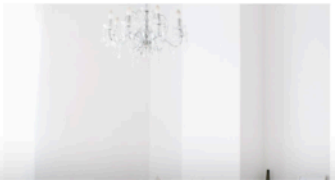
From Figma wireframes to Figma Prototype (low-fidelity & high-fidelity)


app
Ask to edit
Share
12%
Page 1
注册栏目
登陆栏目
剧本下拉
地区下拉
场馆下拉
mobile剧本下拉
mobile地区下拉
mobile场馆下拉
mobile预约成功
mobile约本系统
预约成功
约本系统
mobile剧本大页
mobile活动大页
mobile活动详情
mobile剧本详情
mobile场馆详情
mobile场馆大页
mobile注册小栏目
mobile约本
mobile讨论
mobile交流
mobile用户主页
mobile登陆小栏目
mobile登陆
注册
登陆
关于
主页
场馆大页
场馆详情
活动大页
剧本大页
活动详情
剧本详情
Comment
Inspect
Export
Search
Give feedback, ask a question, or just leave a note of appreciation. Click anywhere in the file to leave a comment.

主页 场馆 剧本 活动 交流 关于 我的

热门场馆：

查看更多



剧本名: 收获日

简介: 在1982年的美国洛娃州，剥削泛滥，正是资本横行野蛮生长的时期。 机遇和风险...

类型: 机制/欢乐/GTA/黑帮

作者: 出山工作室

查看详情

剧本名: 明日星辰

简介: 故事的背景很宏伟，讲述的是未来的故事，但是描写的场景和现代几乎没有区别...

类型: 科幻/还原/情感

作者: 马里旺

查看详情

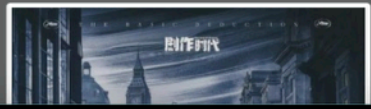
剧本名: 狗子看你不顺眼，猫子也是

简介: 这是一个偏童话风的剧本，虽然大家都是宠物，但也可以和人类交流，可以做一...

类型: 欢乐/机制/沉浸

作者: 凡几剧制

查看详情



剧本名: 收获日

简介: 在1982年的美国洛娃州，剥削泛滥
是资本横行野蛮生长的时期。 机遇和风

类型: 机制/欢乐/GTA/黑帮

作者: 出山工作室

查看详情

本名: 狗子看你不顺眼，猫子也是

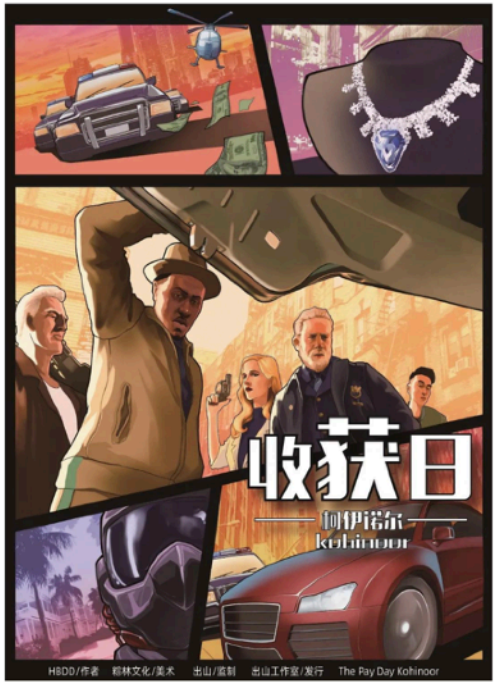
介: 这是一个偏童话风的剧本，虽然大家都
宠物，但也可以和人类交流，可以做一...

型: 欢乐/机制/沉浸

者: 凡几剧制

查看详情

收获日



在1982年的美国洛娃州，剥削泛滥，正是资本横行野蛮生长的时期。 机遇和风险同在，金钱与权欲当先。 初涉黑道的新丁、金盆洗手的恶魔，谁能抹掉身上的罪孽，重做选择？ 一颗名为柯伊诺尔的珍贵宝石将所有人卷入了各个黑道组织、犯罪集团间的恩怨纠葛之中。

关闭

喷火龙剧本杀推理社 所有场馆 重设

MUA
• MUA • いらっしゃいませ •
目馬推理社

藏狐推理馆
ZANGHUTUILIGUAN

逐鹿
ZHULU

场馆:1

场馆名: 目马推理社

评分(满分50）: 50.0

人均价格: 71.0

区域:大学城

查看详情

场馆:2

场馆名: 藏狐剧本推理社

评分(满分50）: 40.0

人均价格: 74.0

区域:车站大道

查看详情

场馆:3

场馆名: 逐鹿沉浸式实景探案馆

评分(满分50）: 45.0

人均价格: 106.0

区域:五马街/大南门

查看详情

主页 剧本 场馆 约本 个人 关于 登出

场馆:1

场馆名: 目马推理社

评分(满分50）：50.0

人均价格: 71.0

区域:大学城

查看详情

馆:3

馆名: 逐鹿沉浸式实景探案馆

分(满分50）：45.0

均价格: 106.0

域:五马街/大南门

查看详情

玖月 推理 探案馆 JIUYUETUILI

白日梦想家

WHO AM I

目马推理社

• MUA • いらっしゃいませ •

目馬推理社

近期活动：98元 「城限剧本杀」新人券

详细地址：勤业路学府印象23132314室

联系方式：18815135510

关闭

开新车

"车"名	详情描述	最新回复
欢迎玩家进行任何测评、评价、约本	*不写条评论不合适吧？*	29 November, 2021 评论
阵营本折迁测评！明灯推理社速来	*已经凑满5人了，只需要两个男孩子！下周一19：00！*	26 November, 2021 评论
大学城这边想约本！独家本带测评！	*xx剧本杀馆最新的剧本真的很厉害，不测不合适吧！本周末约一波？*	08 December, 2021 评论
本周末欧洲城急速拼车！	*只要是情感本都可以！目前有两个人急速拼车！*	12 December, 2021 评论
家人们想要一起约千千晚星	*独家剧本欸！！所有人都可以来拼，十二月寒假我都可以约！*	12 December, 2021 评论

小贴士

为建立友好社区，请各位玩家文明用词，和谐相处，成为彼此的朋友！

约本区

- 大学城上车！
- 市府路速来！
- 米房就等你了！
- 欧洲城不来不合适吧！
- 五马街差你一个！

独家预告

主页 剧本 场馆 约本 个人 关于 登出

欢迎玩家进行任何测评、评价、约本

阵营本拆迁测评！明灯推理社速来

大学城这边想约本！独家本带测评！

本周末欧洲城急速拼车！

家人们想要一起约千千晚星

预约

剧本名	场馆名	日期
拆迁	挺好的推理社	202
甜蜜蜜	甜蜜蜜场馆	202
千千阙歌	欧洲城	202
都挺好	市府路	202
千千晚星	八月推理探案馆	202

八月推理探案馆独家剧本

人数：3女5男

时长：4-4.5小时

1994年9月2日，这一天，你赴约参加了一场高中同学聚会，十年了，你们各自的生活都产生了巨大的变化。下班以后，你们在电视台大楼碰了头，仍有一位同学迟迟未到场，他是谁？

徐徐回望，曾属于彼此的晚上……红红仍是你，赠我的心中艳阳……伴随着优美的音乐广播与离奇死亡案件，你们的眼中出现了种种光怪陆离的，诡谲怪诞的场景……

这个剧本实在是太值得约本了，家人们冲起来呀！

关闭

本区

大学城上车！

市府路速来！

米房就等你了！

欧洲城不来不合适吧！

五马街差你一个！

家预告

月推理探案馆独家剧本

化/还原/微情感

月星光影"的第一部，开端之作

一起来约本吧！

详情

预约

主页 剧本 场馆 约本 个人 关于 登出

欢迎玩家进行任何测评、评价、约本

阵营本拆迁测评！明灯推理社速来

大学城这边想约本！独家本带测评！

本周末欧洲城急速拼车！

家人们想要一起约千千晚星

预约

剧本名	场馆名	日期			
拆迁	挺好的推理社	2021/11/29	19:00	5	2
甜蜜蜜	甜蜜蜜场馆	2021/11/28	21：00	5	3
千千阙歌	欧洲城	2021/12/13	10:00	4人	2人
都挺好	市府路	2021/12/14	11：00	4人	3人
千千晚星	八月推理探案馆	2021/12/30	13：00	2人	3人

×

预填信息

请输入预约剧本

请输入预约的场馆名+区域

请输入预约日期(例:2021/11/10)

请输入到场时间(例:10:00)

请输入预约人数(例:5人)

请输入所缺人数(例:2人)

请输入您的联系方式，我们将尽快通知您是否拼车成功。

提交预约

关闭

区

学城上车！
府路速来！
房就等你了！
洲城不来不合适吧！
马街差你一个！

预告

八月推理探案馆独家剧本
现代/还原/微情感
“日月星光影”的第一部，开端之作
一起来约本吧！

详情 预约

主页 剧本 场馆 约本 个人 关于 登出

欢迎来到您的主页

昵称: 111111
用户名: 111
用户Id: 7
邮箱: 111@qq.com
性别: 女
年龄: 22
玩过的剧本: 无
想要测评的剧本: 无
喜欢的剧本类型: 言情
活动区域: 香港
加入会员日期:

更新个人信息：

性别:

女

年龄:

22

玩过的剧本:

无

想要测评的剧本:

无

喜欢的剧本类型:

言情

活动区域:

香港

我的预约

剧本名	场馆名	日期	时间	已有人数	等待人数
拆迁	*挺好的推理社*	*2021/11/29*	*19:00*	*5*	*2*

主页 剧本 场馆 约本 个人 关于 登出

查找场馆
场馆入驻

入驻后，我们将为您提供以下服务：

提升场馆曝光	组织约本活动	宣传独家剧本
私人定制场馆的展示形式和信息，调整场馆曝光位置，帮助您让更多的爱好者看到。	您可以定期提交场馆的拼车需求，我们将帮助您在讨论区组织约本，帮助您获得新的客源。	您的独家剧本将被投放至广告位，提高独家剧本的曝光量，展示场馆的独特风采。

实现入驻，只需要三步。

1.在以下表格填写基础信息	2.在详细信息区域填写门店信息（如：地理位置、营业执照、法人身份证等）	3.提交后审核预计1-3工作日，我们的工作人员将会通过您预留的联系方式与您联系沟通入驻事项。

填写表格，立即入驻。

场馆名

请输入场馆名

邮箱

请输入邮箱

联系电话

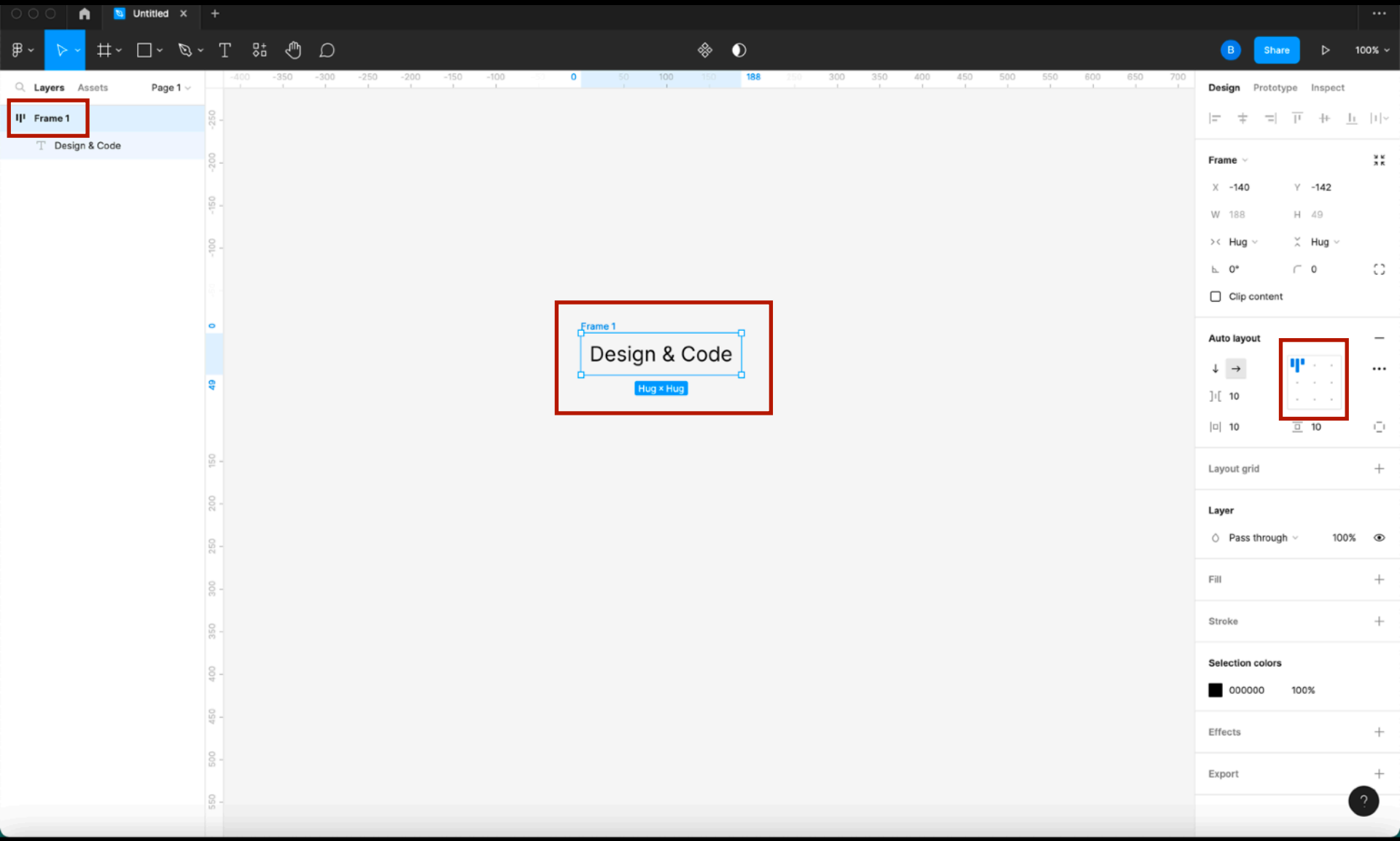
请输入联系电话

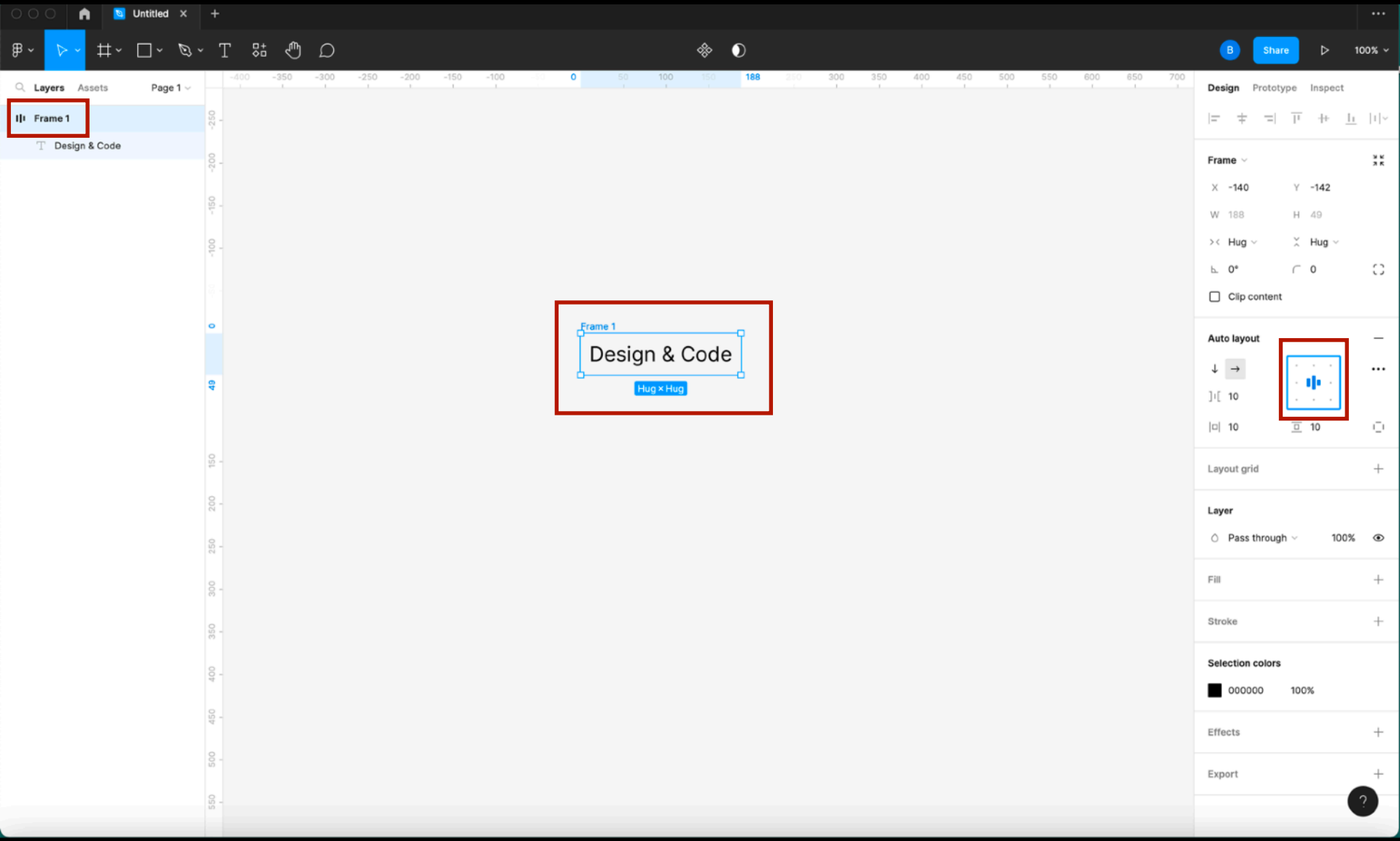
详细信息

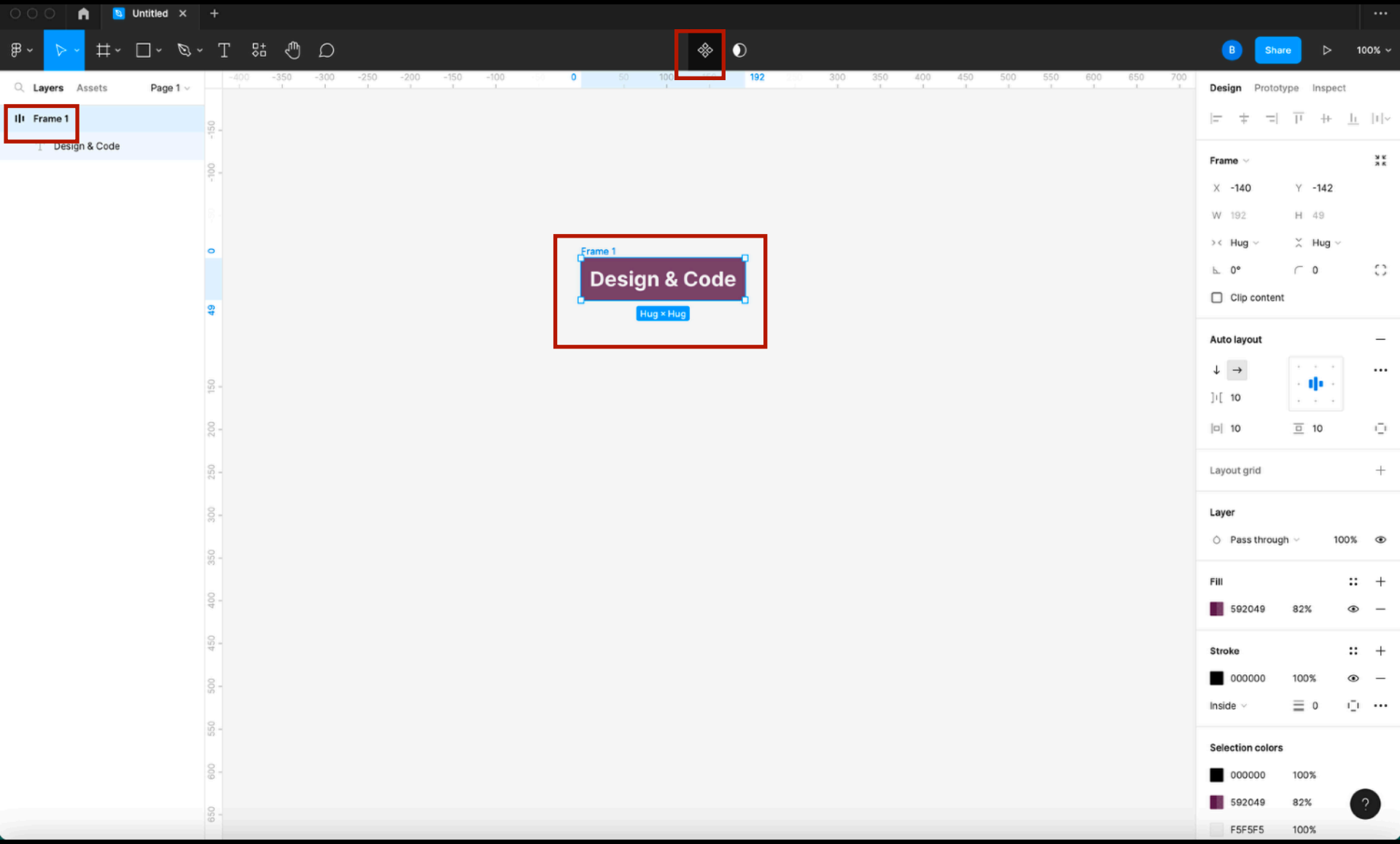
填写场馆的详细信息和主要诉求，我们将会尽快联系您。

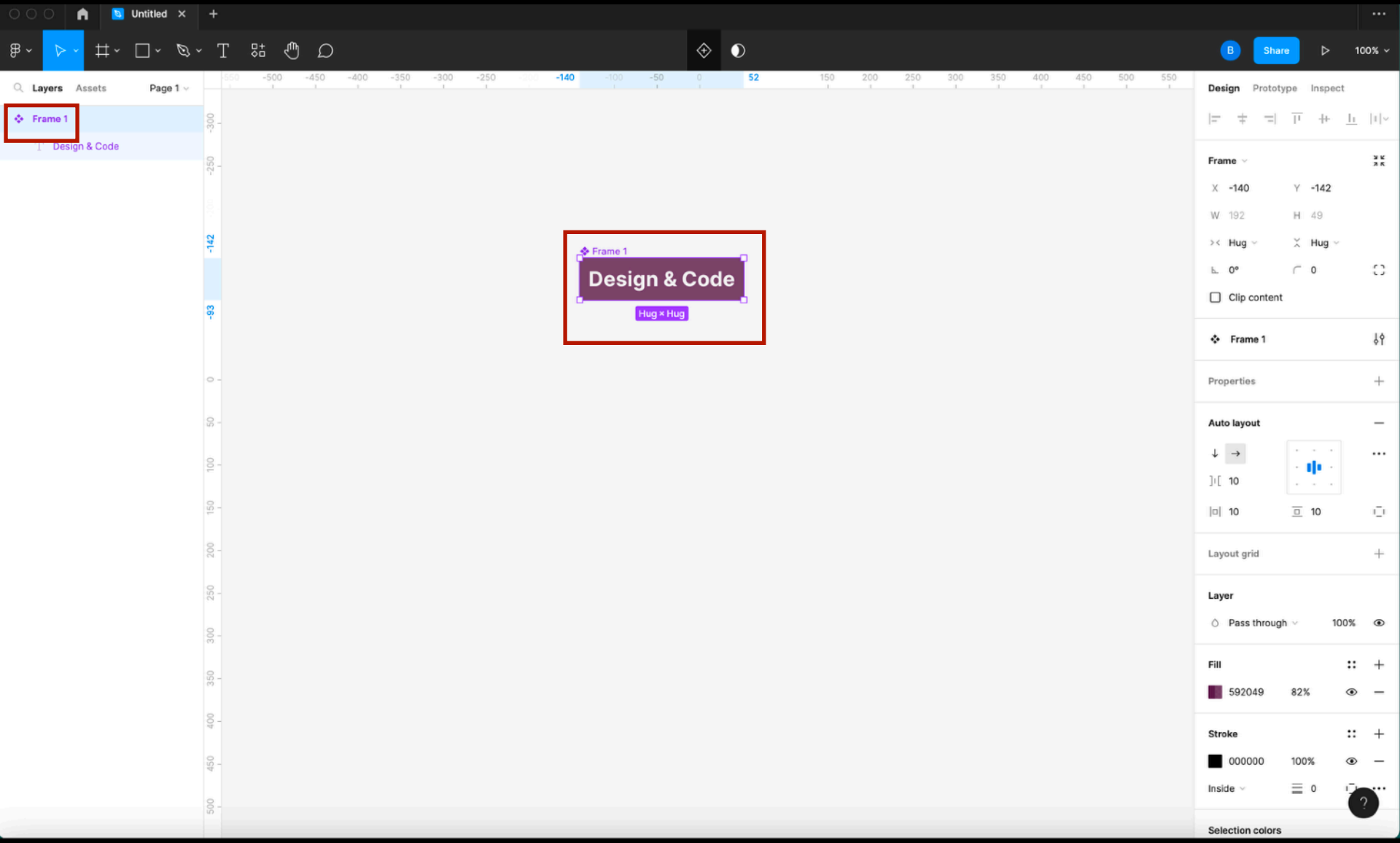
From Figma wireframes to HTML templates.

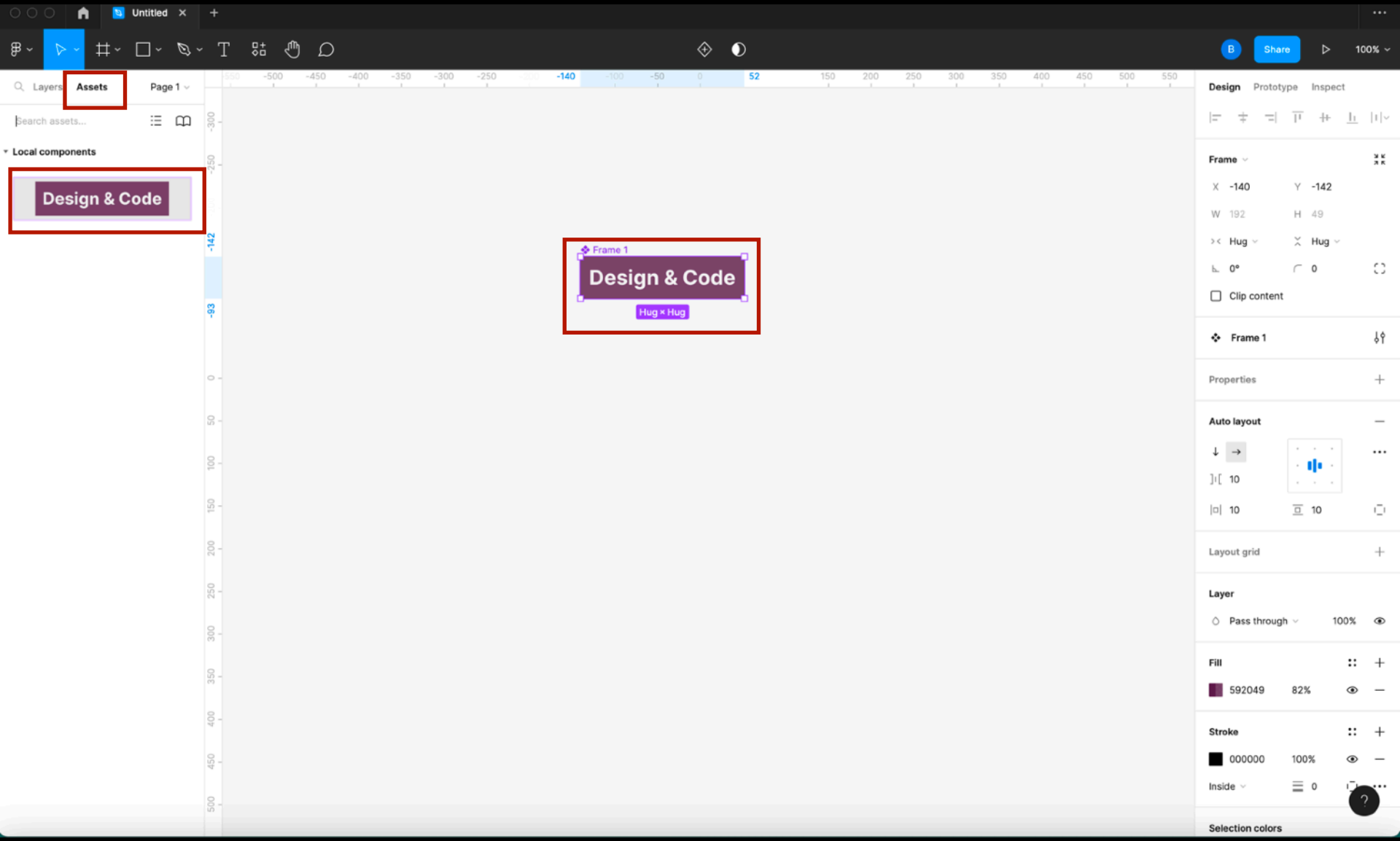
1. Auto-layout component.
2. Sharing of component across files.
3. Building wireframe screens with components.
4. Using plugins.
5. Generate HTML and CSS.


Untitled
Share
100%
Layers
Assets
Page 1
Frame 1
Design & Code
Frame 1
Design & Code
Hug × Hug
Design
Prototype
Inspect
Frame
X -140
Y -142
W 188
H 49
Hug
Hug
0°
0
Clip content
Auto layout
10
10
10
Layout grid
Layer
Pass through
100%
Fill
Stroke
Selection colors
000000
100%
Effects
Export

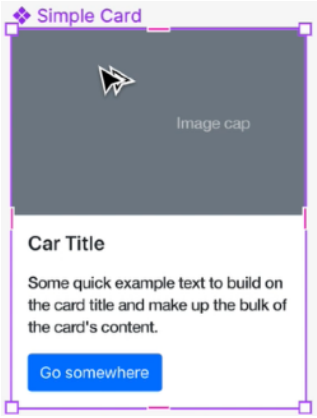

Untitled
Share
100%
Layers
Assets
Page 1
Frame 1
Design & Code
Frame 1
Design & Code
Hug × Hug
Design
Prototype
Inspect
Frame
X -140
Y -142
W 188
H 49
Hug
Hug
0°
0
Clip content
Auto layout
10
10
10
Layout grid
Layer
Pass through
100%
Fill
Stroke
Selection colors
000000
100%
Effects
Export


Untitled
Share
Layers
Assets
Page 1
Frame 1
Design & Code
Frame 1
Design & Code
Hug × Hug
Design
Prototype
Inspect
Frame
X -140
Y -142
W 192
H 49
Hug
Hug
0°
0
Clip content
Auto layout
10
10
10
Layout grid
Layer
Pass through
100%
Fill
592049
82%
Stroke
000000
100%
Inside
0
Selection colors
000000
100%
592049
82%
F5F5F5
100%

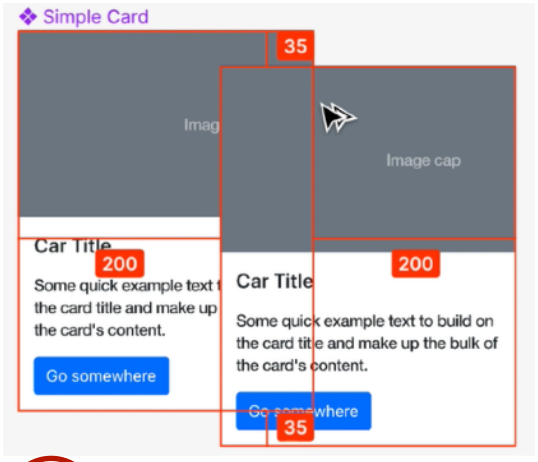

Untitled
Share
100%
Layers
Assets
Page 1
Frame 1
Design & Code
Frame 1
Design & Code
Hug × Hug
Design
Prototype
Inspect
Frame
X -140
Y -142
W 192
H 49
Hug
Hug
0°
0
Clip content
Frame 1
Properties
Auto layout
10
10
10
Layout grid
Layer
Pass through
100%
Fill
592049
82%
Stroke
000000
100%
Inside
0
Selection colors


Untitled
Share
100%
Layers
Assets
Page 1
Search assets...
Local components
Design & Code
Frame 1
Design & Code
Hug × Hug
Design
Prototype
Inspect
Frame
X -140
Y -142
W 192
H 49
Hug
Hug
0°
0
Clip content
Frame 1
Properties
Auto layout
10
10
10
Layout grid
Layer
Pass through
100%
Fill
592049
82%
Stroke
000000
100%
Inside
0
Selection colors

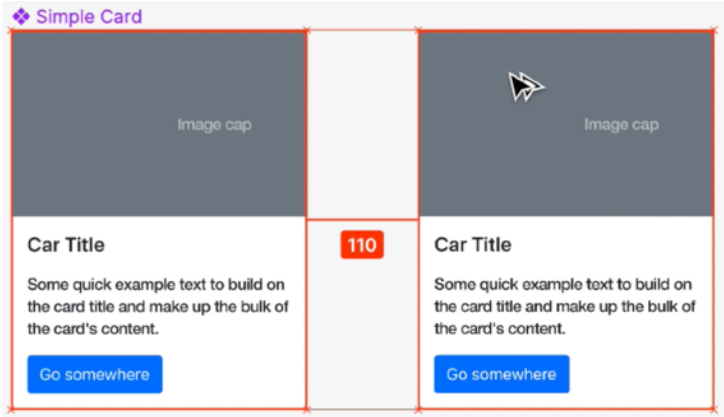
Create an instance from a master component.



1



2



3

Text

Font
Weight
Size
Line Height
Letter Spacing
Paragraph Spacing
Indentation

Color

Fill
Stroke
Background Color
Opacity.

Effects

Drop Shadow
Inner Shadow
Blur

Component features that can be overwritten by the instance.

Guide to libraries in Figma

Sharing of component across files

Who can use this feature

Libraries are available on any plan.

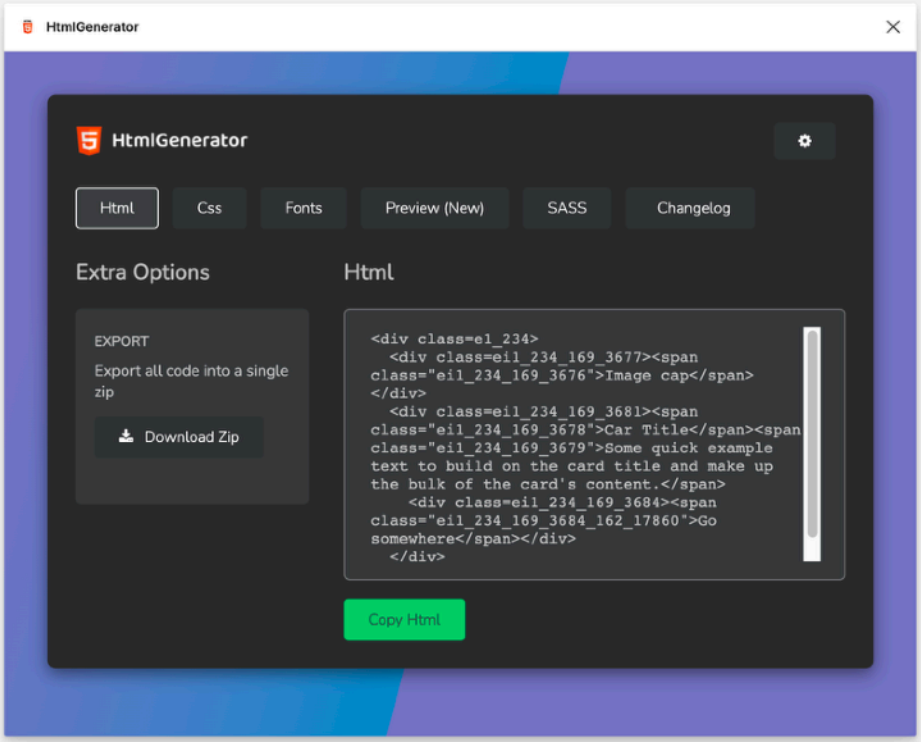
Publishing components to a library is available on the Professional, Organization and Enterprise plan.

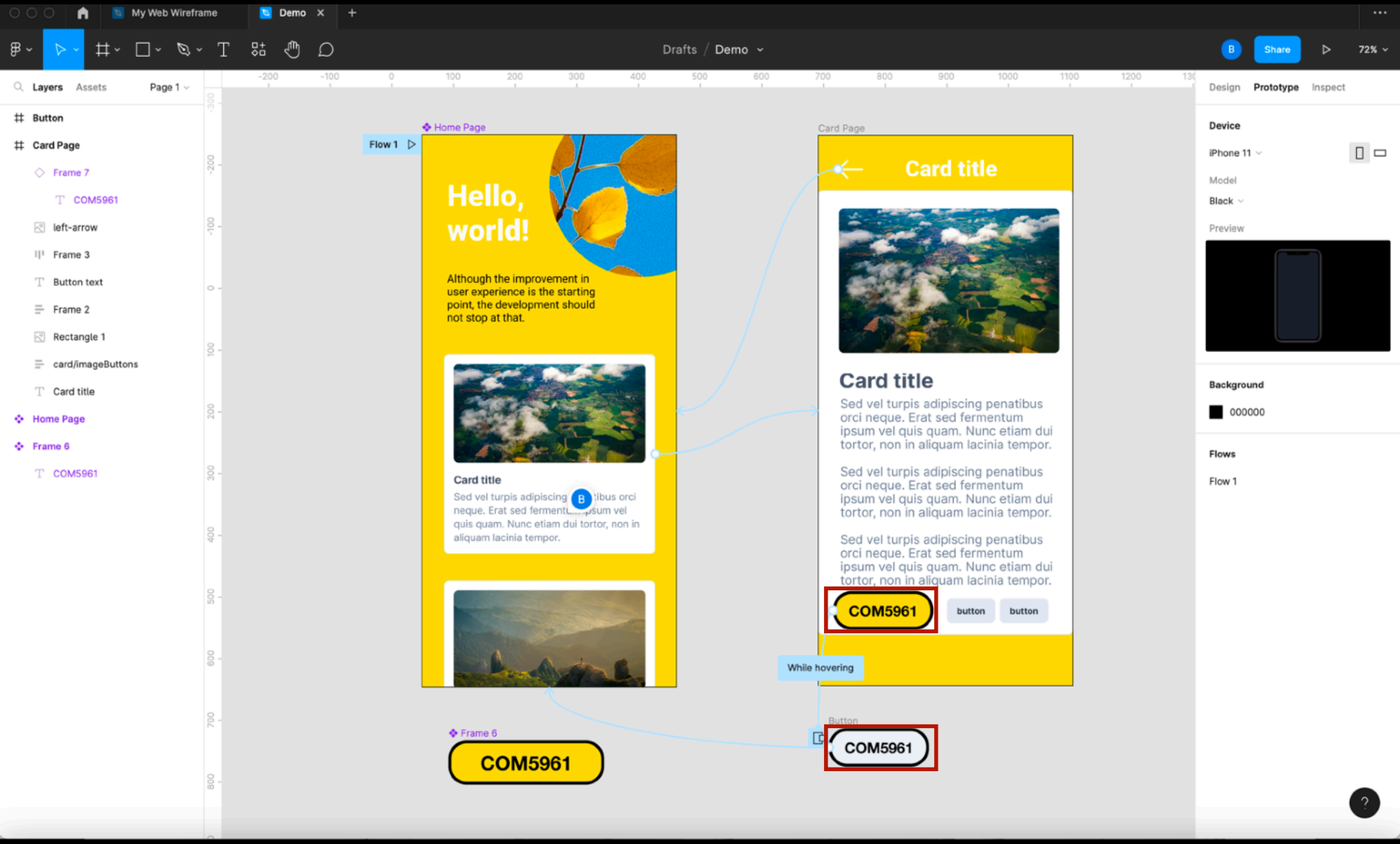
Anyone with `can edit` permission to a file can publish styles and components to a library.

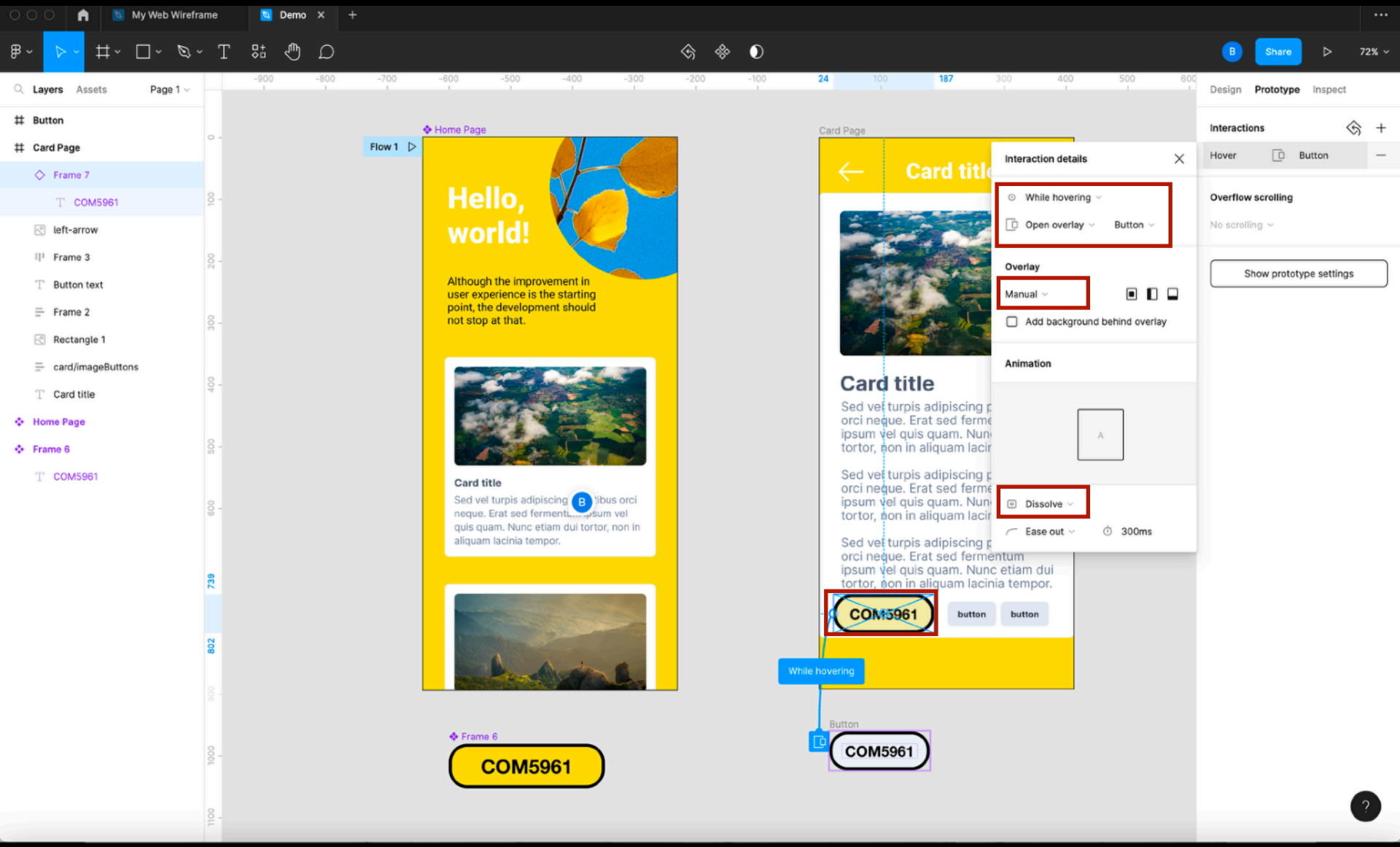
Useful plugins for components creation

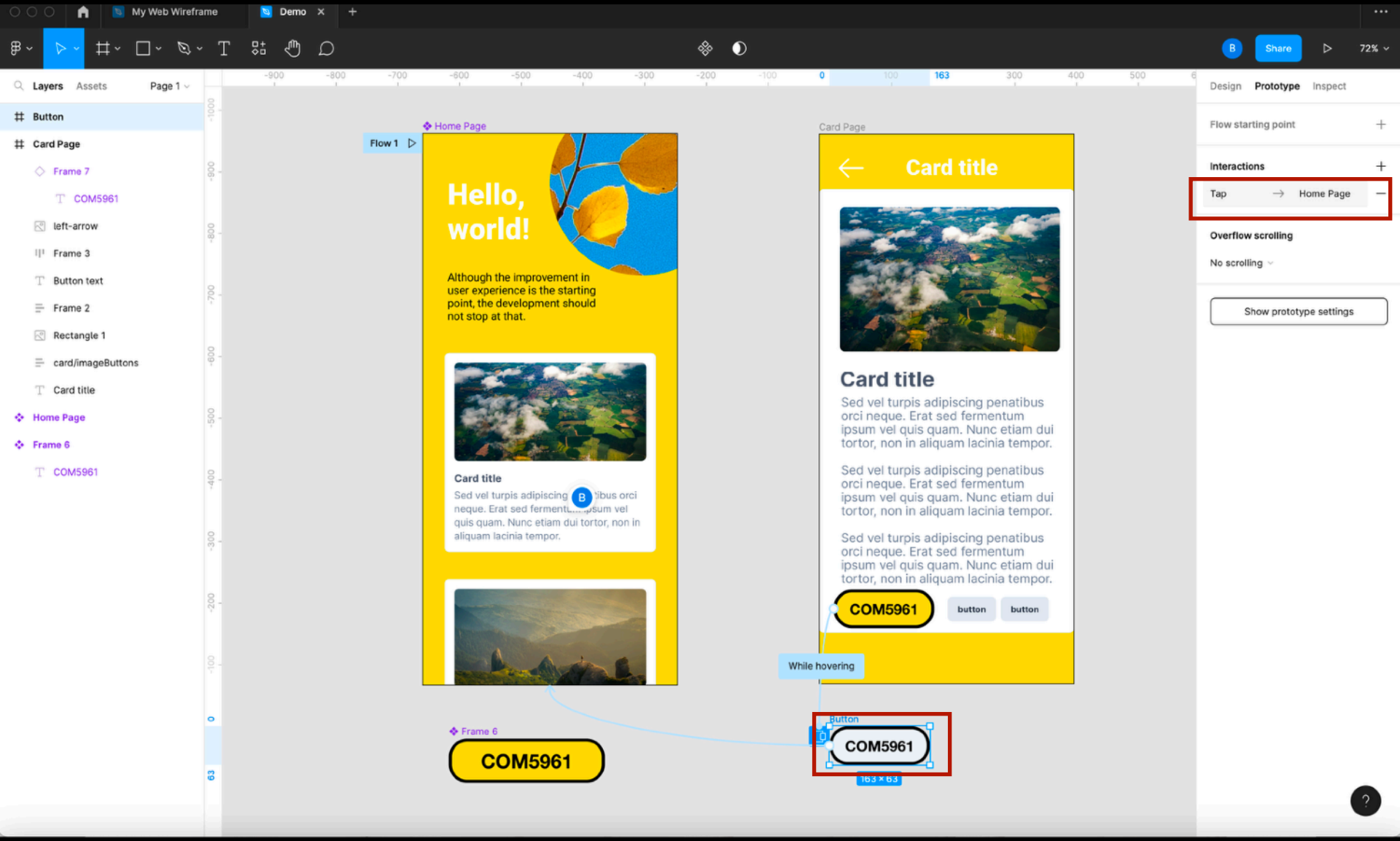
1. Font Awesome Icons
2. LORE (for Loren Ipsum generation)
3. Unsplash
4. HtmlGenerator

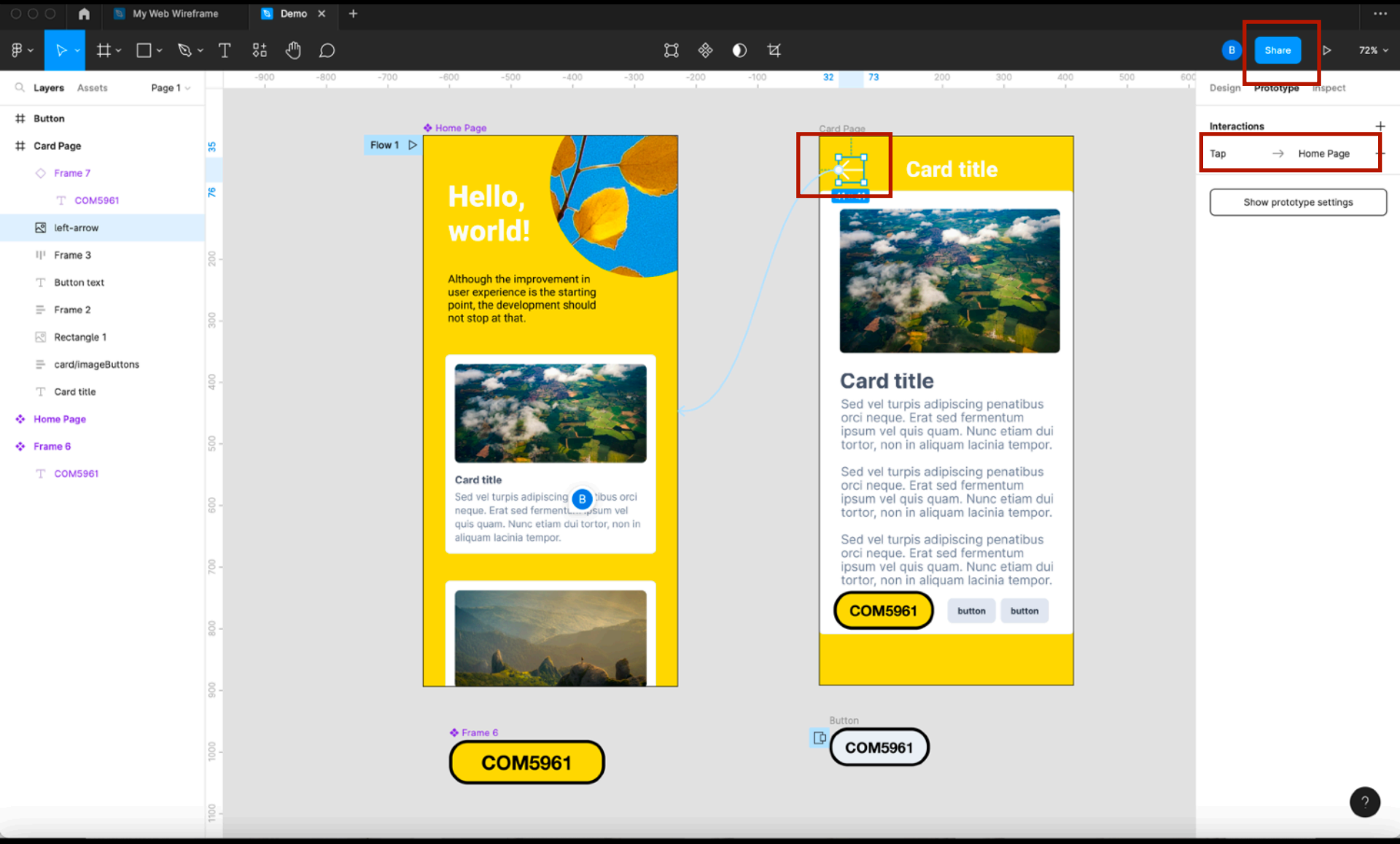
HTML/CSS code generation

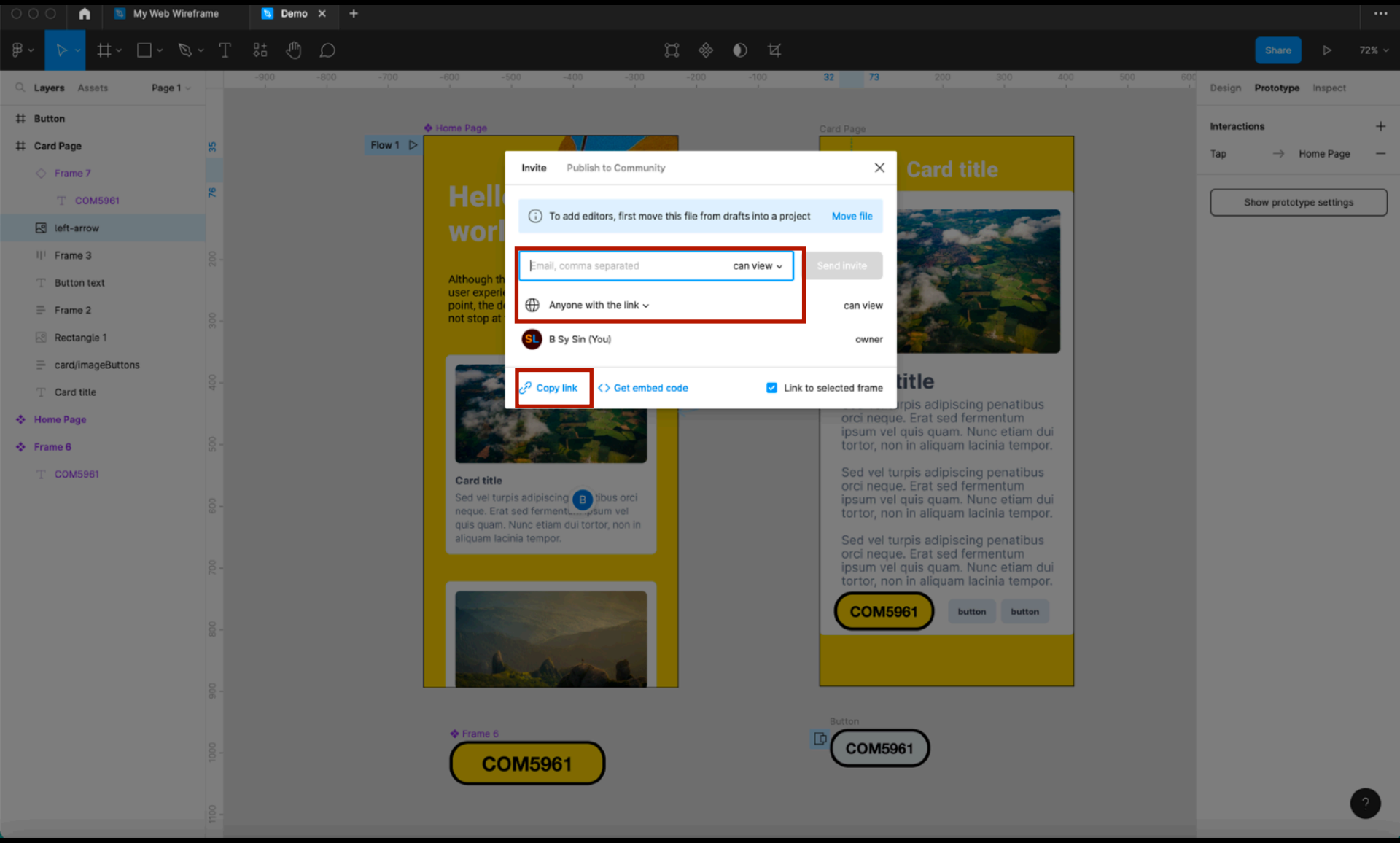



My Web Wireframe
Demo
Drafts / Demo
Share
72%
Layers
Assets
Page 1
Button
Card Page
Frame 7
COM5961
left-arrow
Frame 3
Button text
Frame 2
Rectangle 1
card/imageButtons
Card title
Home Page
Frame 6
COM5961
Design
Prototype
Inspect
Device
iPhone 11
Model
Black
Preview
Background
000000
Flows
Flow 1
Flow 1
Home Page
Hello, world!
Although the improvement in user experience is the starting point, the development should not stop at that.
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Card Page
Card title
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
COM5961
button
button
While hovering
Button
COM5961
Frame 6
COM5961

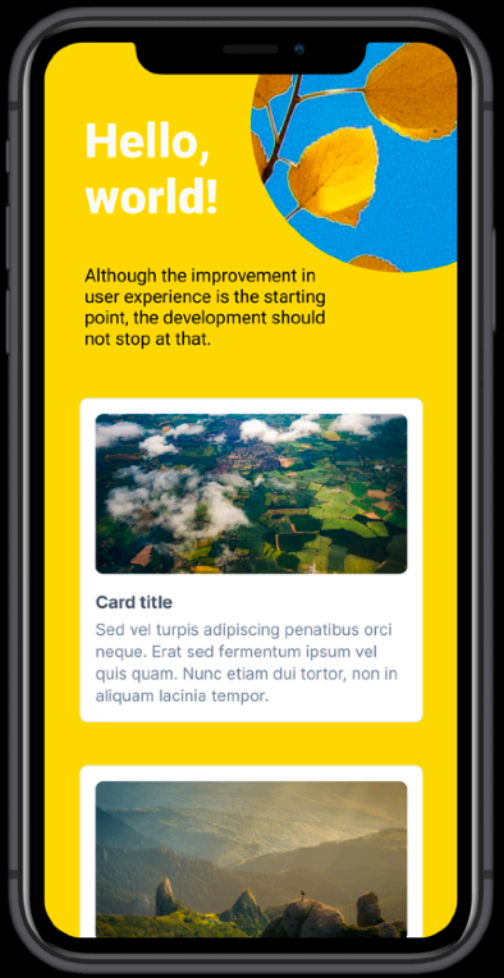

My Web Wireframe
Demo
Share
72%
Layers
Assets
Page 1
Button
Card Page
Frame 7
COM5961
left-arrow
Frame 3
Button text
Frame 2
Rectangle 1
card/imageButtons
Card title
Home Page
Frame 6
COM5961
Home Page
Flow 1
Hello, world!
Although the improvement in user experience is the starting point, the development should not stop at that.
Card title
Sed vel turpis adipiscing
neque. Erat sed fermentum vel
quis quam. Nunc etiam dui tortor, non in
aliquam lacinia tempor.
Frame 6
COM5961
Card Page
Card title
Card title
Sed vel turpis adipiscing
orci neque. Erat sed ferme
ipsum vel quis quam. Nunc
tortor, non in aliquam lacinia tempor.
COM5961
button
button
While hovering
Button
COM5961
Interaction details
While hovering
Open overlay
Button
Overlay
Manual
Add background behind overlay
Animation
Dissolve
Ease out
300ms
Design
Prototype
Inspect
Interactions
Hover
Button
Overflow scrolling
No scrolling
Show prototype settings


My Web Wireframe
Demo
Share
72%
Layers
Assets
Page 1
Button
Card Page
Frame 7
COM5961
left-arrow
Frame 3
Button text
Frame 2
Rectangle 1
card/imageButtons
Card title
Home Page
Frame 6
COM5961
Home Page
Flow 1
Hello, world!
Although the improvement in user experience is the starting point, the development should not stop at that.
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Card Page
Card title
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
COM5961
button
button
While hovering
Frame 6
COM5961
Button
COM5961
163 × 63
Design
Prototype
Inspect
Flow starting point
Interactions
Tap
Home Page
Overflow scrolling
No scrolling
Show prototype settings

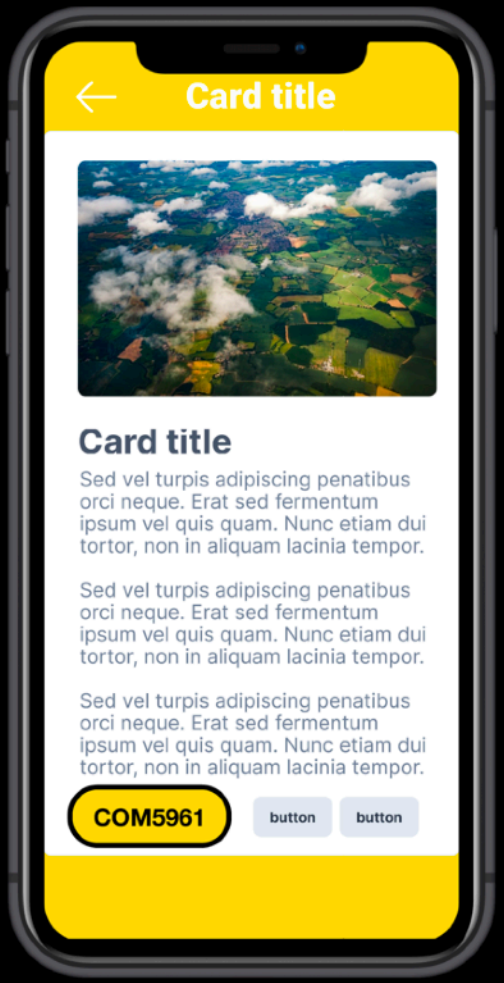

My Web Wireframe
Demo
Share
72%
Layers
Assets
Page 1
Button
Card Page
Frame 7
COM5961
left-arrow
Frame 3
Button text
Frame 2
Rectangle 1
card/imageButtons
Card title
Home Page
Frame 6
COM5961
Design
Prototype
Inspect
Interactions
Tap
Home Page
Show prototype settings
Flow 1
Home Page
Hello, world!
Although the improvement in user experience is the starting point, the development should not stop at that.
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Card Page
Card title
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
COM5961
button
button
Frame 6
COM5961
Button
COM5961


My Web Wireframe
Demo
Share
72%
Layers
Assets
Page 1
Button
Card Page
Frame 7
COM5961
left-arrow
Frame 3
Button text
Frame 2
Rectangle 1
card/imageButtons
Card title
Home Page
Frame 6
COM5961
Design
Prototype
Inspect
Interactions
Tap
Home Page
Show prototype settings
Flow 1
Home Page
Card Page
Invite
Publish to Community
To add editors, first move this file from drafts into a project
Move file
Email, comma separated
can view
Send invite
Anyone with the link
can view
B Sy Sin (You)
owner
Copy link
Get embed code
Link to selected frame
Card title
Card title
Sed vel turpis adipiscing penatibus orci neque. Erat sed fermentum ipsum vel quis quam. Nunc etiam dui tortor, non in aliquam lacinia tempor.
COM5961
button
button
Frame 6
COM5961
Button
COM5961

My Web Wireframe | Demo | Page 1 - Demo



My Web Wireframe | Demo | Page 1 - Demo



Assignment #4

1. Develop future journey map and story map based on your functional and content requirements.
2. Create sorting cards based on story cards for performing closed or open card sort.
3. Develop an information architecture (sitemap) based on card sorting result.
4. Use Figma to design wireframes and clickable prototype based on sitemap structure.
5. Identify key hypothesis to be validated through prototype.
6. Design usability test using the Usability Dashboard.
7. Conduct usability test using Figma prototype and write up your analysis.

Thanks for joining me today!