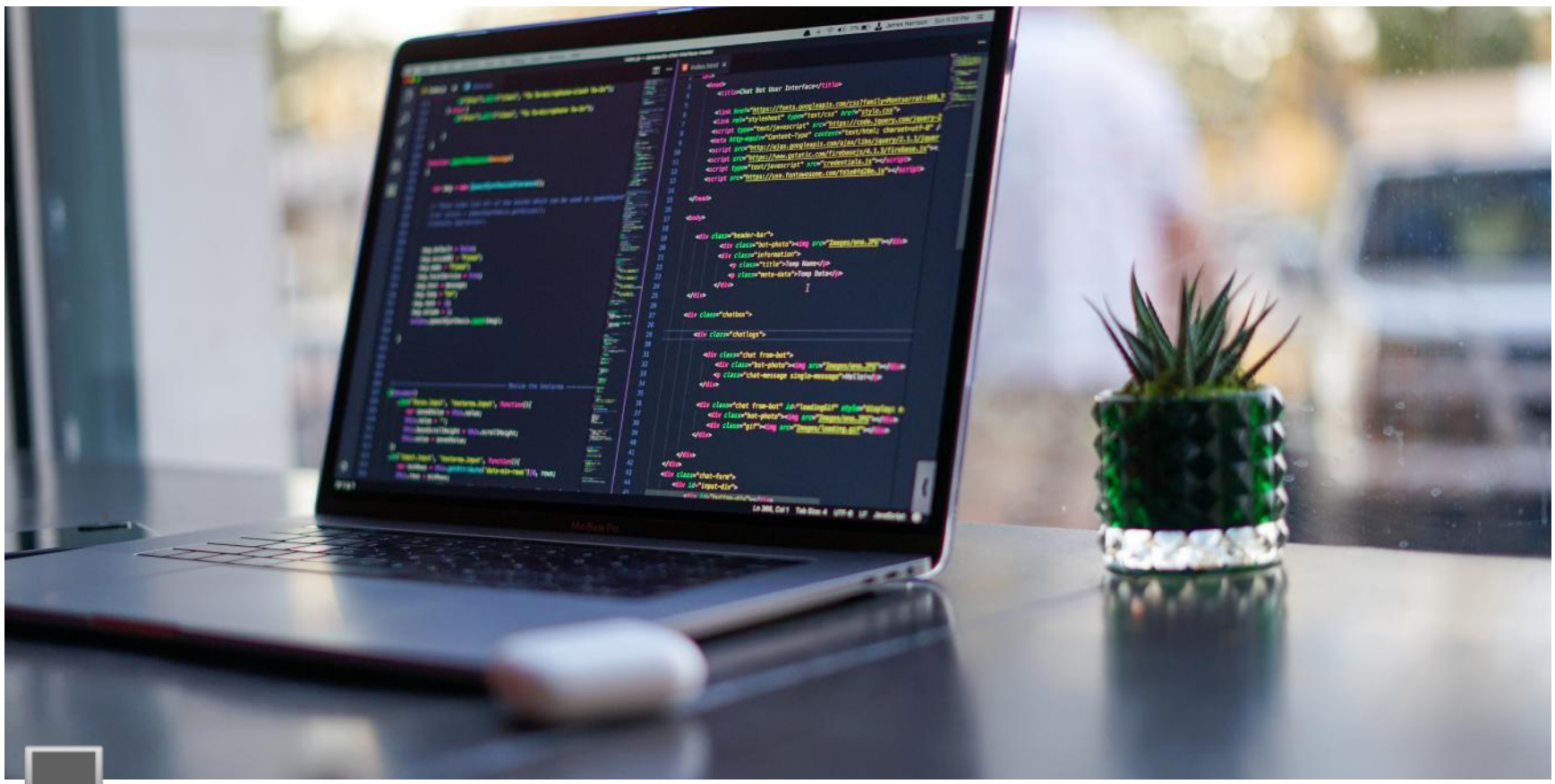


Week 3 Lecture 1

Class	BSCCS2003
Created	@September 18, 2021 5:06 PM
Materials	
Module #	16
Type	Lecture
Week #	3

Overview of MVC

Outline

- MVC paradigm
- Views and User interfaces
- Tools & Techniques
- Accessibility

Model: Store e-mails on the server; indexed and ready to manipulate

View: Display a list of e-mails; Read individual e-mail

Controller: Sort e-mails; delete; archive (Usually the business logic of an app)

Model-View-Controller

- Origins: Smalltalk-80
- Separation of responsibilities - **Abstraction**
- Roots in Object-Oriented GUI development

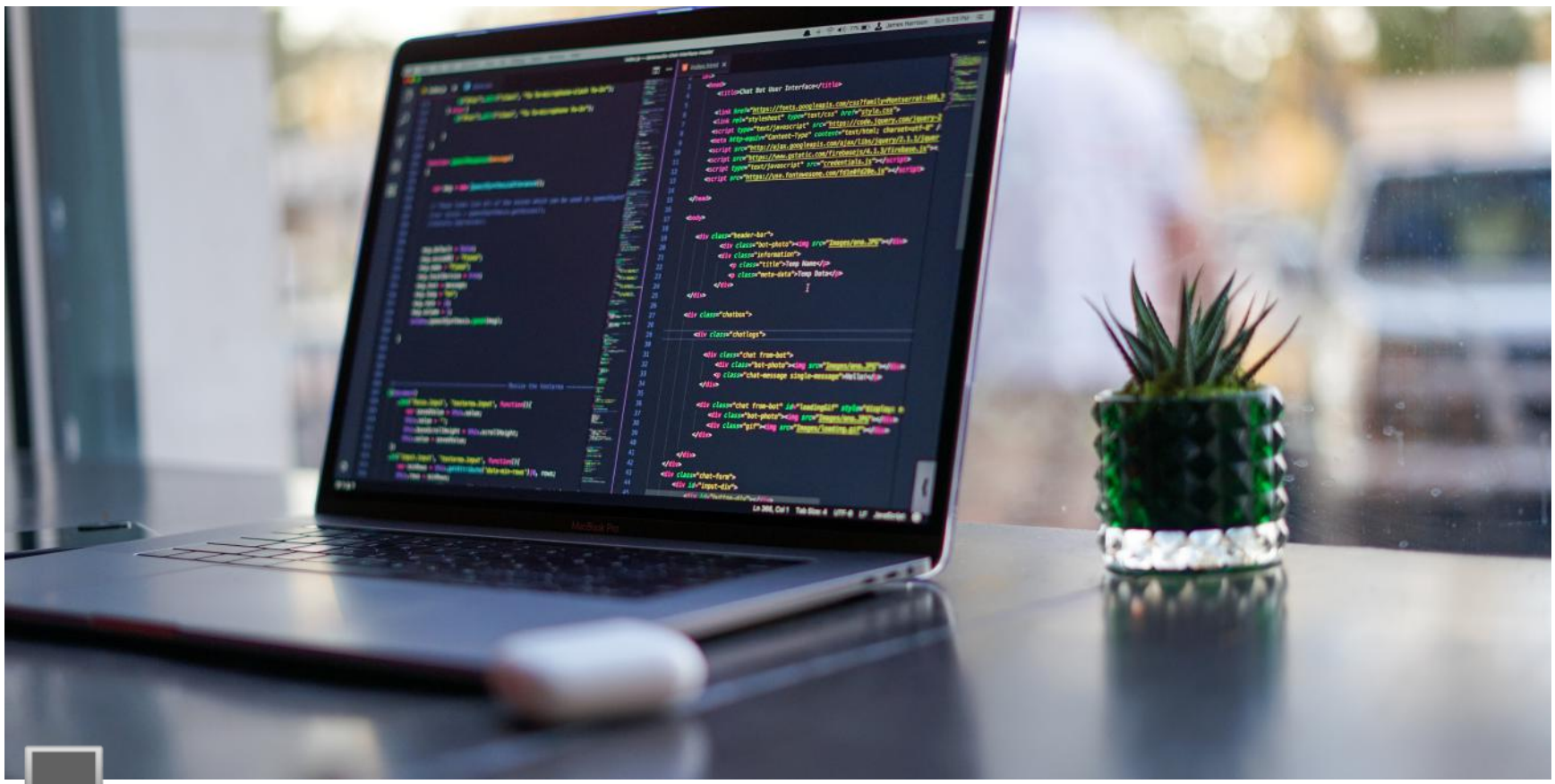
Design Patterns

- Common software patterns
- **Model:** Application object
- **View:** Screen representation
- **Controller:** How does the user interface reacts to user input

Running Example

Student Gradebook

- Input data: for **Model**
  - Student list
  - Course list
  - Student-Course marks
- Outputs: for **View**
  - Marks for individual student
  - Summary of the course
  - Histograms
- Modifications: for **Controller**
  - Add new students
  - Add new courses
  - Modify the marks in a course



Week 3 Lecture 2

Class	BSCCS2003
Created	@September 18, 2021 5:20 PM
Materials	
Module #	17
Type	Lecture
Week #	3

Views

User Interface Design

User Interface

- Screen
- Audio
- Vibration (haptics)
- Motor (door open / close)

User Interaction

- Keyboard / Mouse
- Touchscreen
- Spoken voice
- Custom buttons

User interaction is ...

- determined by the hardware constraints
- different based on various types of devices
- user-agent information useful to identify context

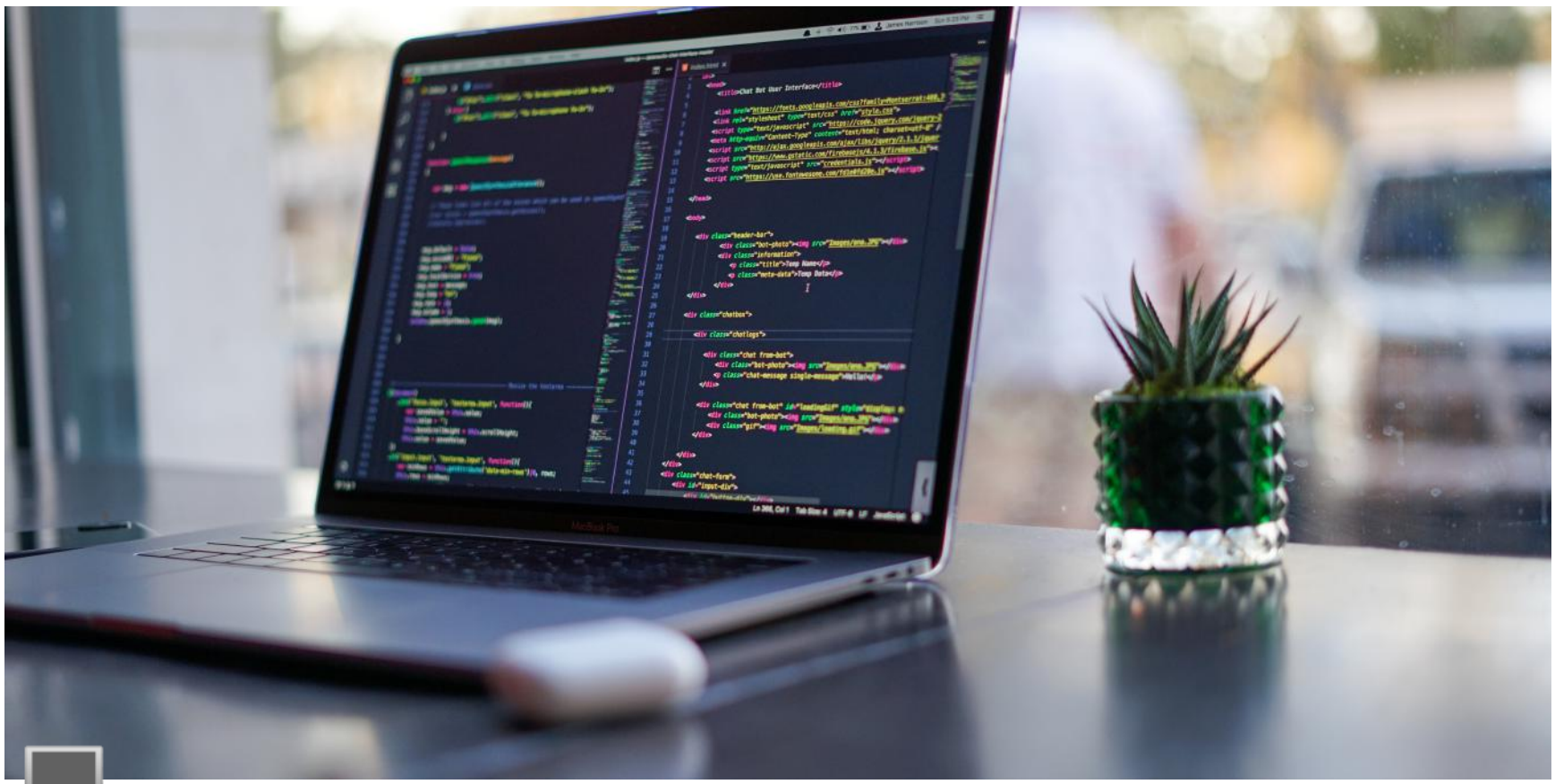
Types of Views

- Fully static
- Partly dynamic
- Mostly dynamic

Output

- HTML - most commonly used - direct rendering
- Dynamic images
- JSON / XML - machine readable

View - any "representation" that is useful to another entity



Week 3 Lecture 3

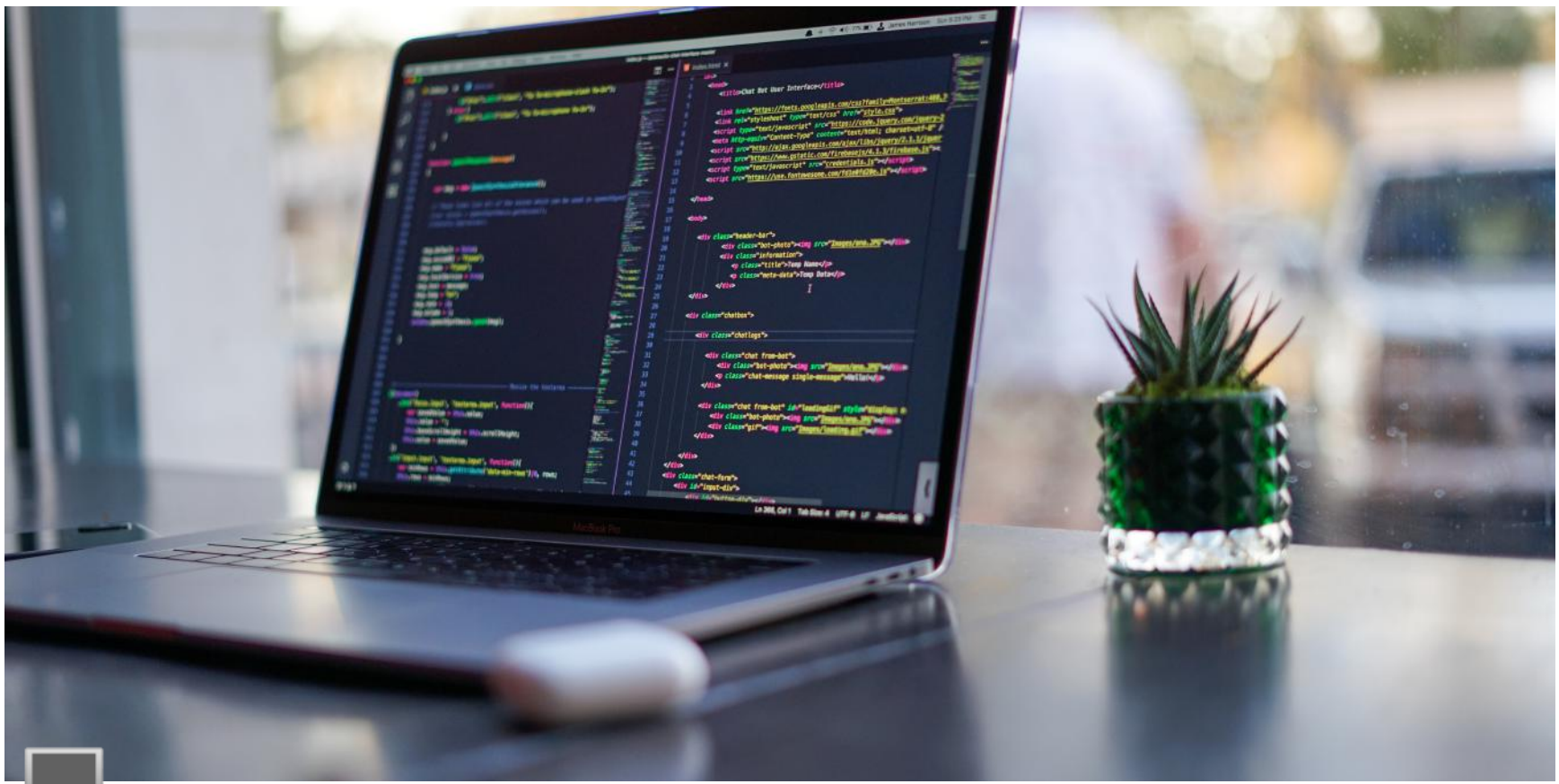
Class	BSCCS2003
Created	@September 18, 2021 5:37 PM
Materials	
Module #	18
Type	Lecture
Week #	3

User Interface Design

- Design for interaction with the user
- Goals:
  - **Simple:** Easy for the user to understand and use
  - **Efficient:** User achieves their goal with minimal effort
- Aesthetics
  - Something which is pleasing to the eyes and looks nice
- Accessibility

Systematic Process

- Functionality requirements gathering - What is needed?
- User and Task analysis - user preferences, task needs
- Prototyping - wireframes, mock-ups
- Testing - User acceptance, usability, accessibility



Week 3 Lecture 4

Class	BSCCS2003
Created	@September 18, 2021 6:27 PM
Materials	
Module #	19
Type	Lecture
Week #	3

Usability Heuristics

Guidelines / Heuristics

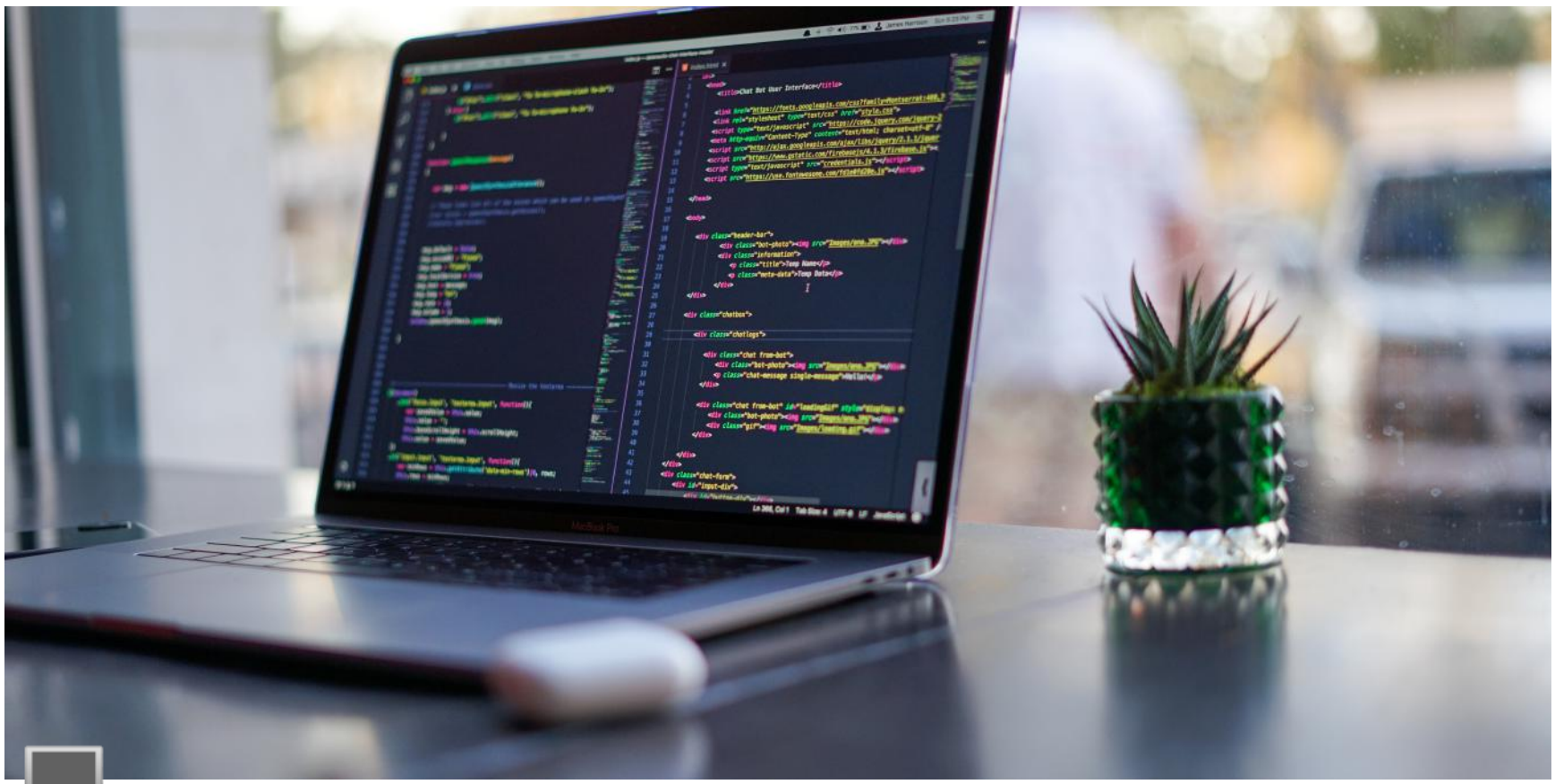
Jakob Nielsen's heuristics for design

https://www.nngroup.com/articles/ten-usability-heuristics/

- Not specific to web apps, or even software UI design
- But it is very useful and relevant

General principles

- Consistency
- Simple and minimal steps
- Simple language
- Minimal and aesthetically pleasing



Week 3 Lecture 5

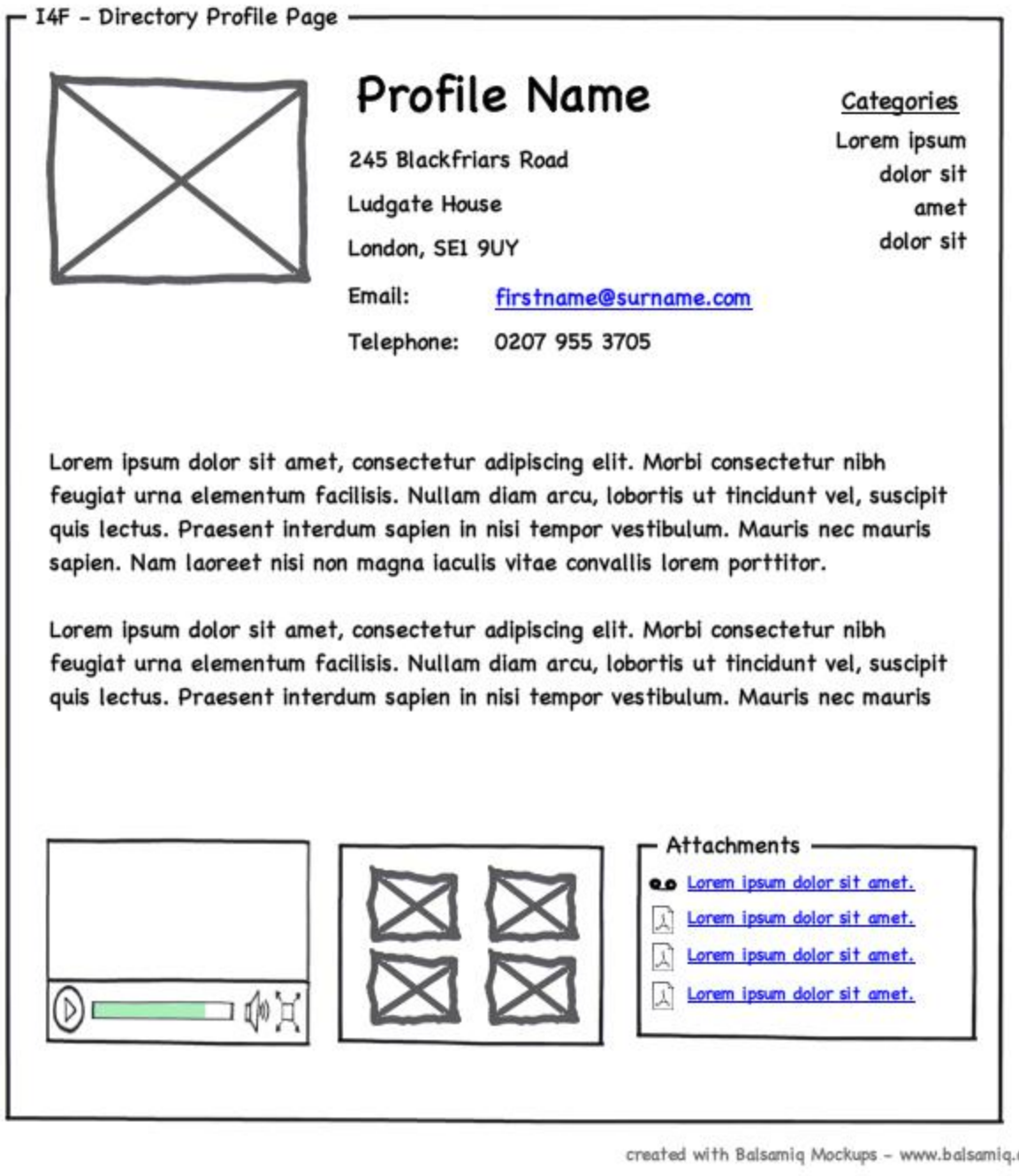
Class	BSCCS2003
Created	@September 18, 2021 6:36 PM
Materials	
Module #	20
Type	Lecture
Week #	3

Tools (part 1)

- Wireframes
- HTML generation
- Templates

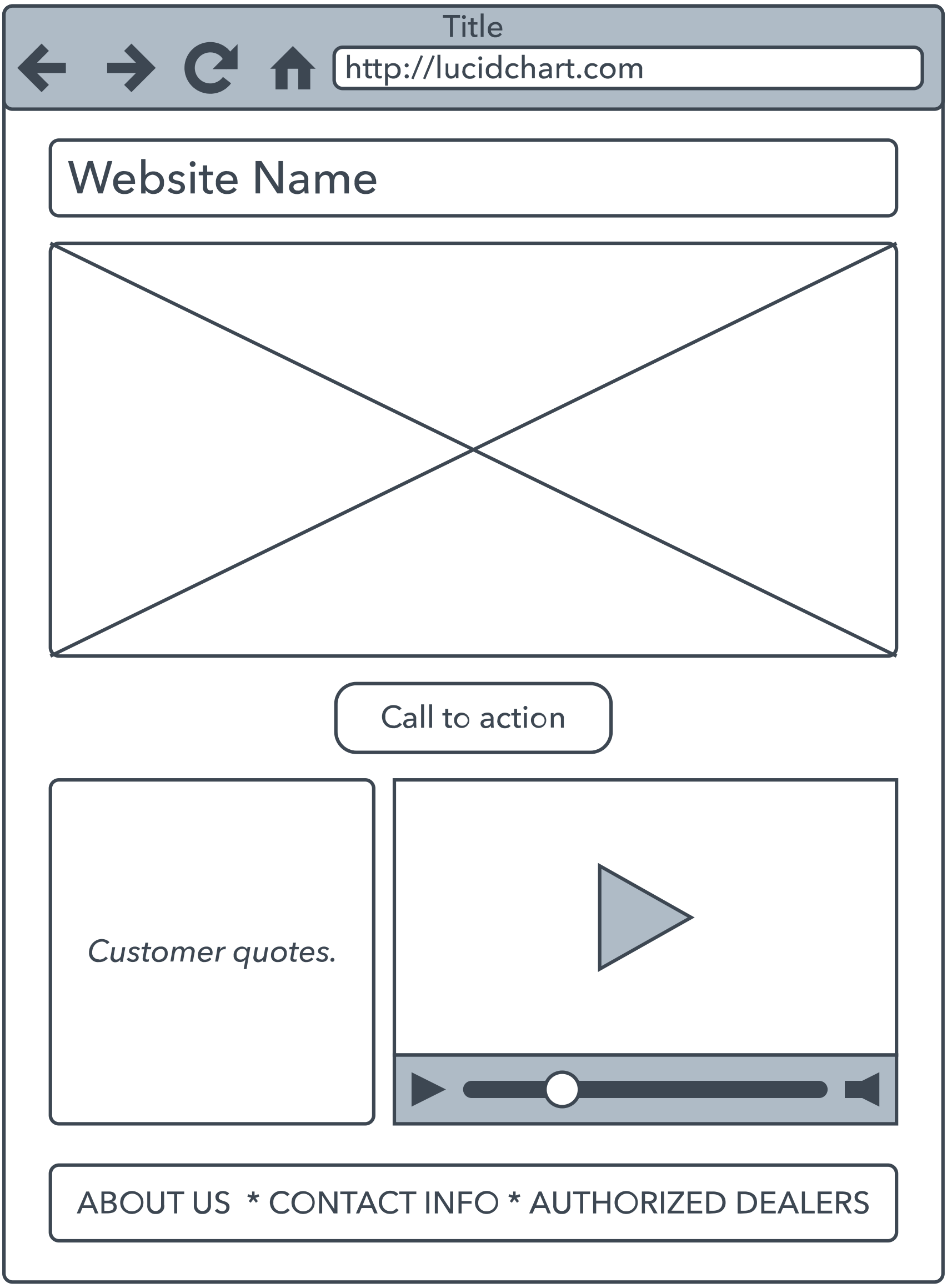
Wireframes

- Visual guide to represent **structure** of the web page
- Information design
- Navigation design
- User interface design

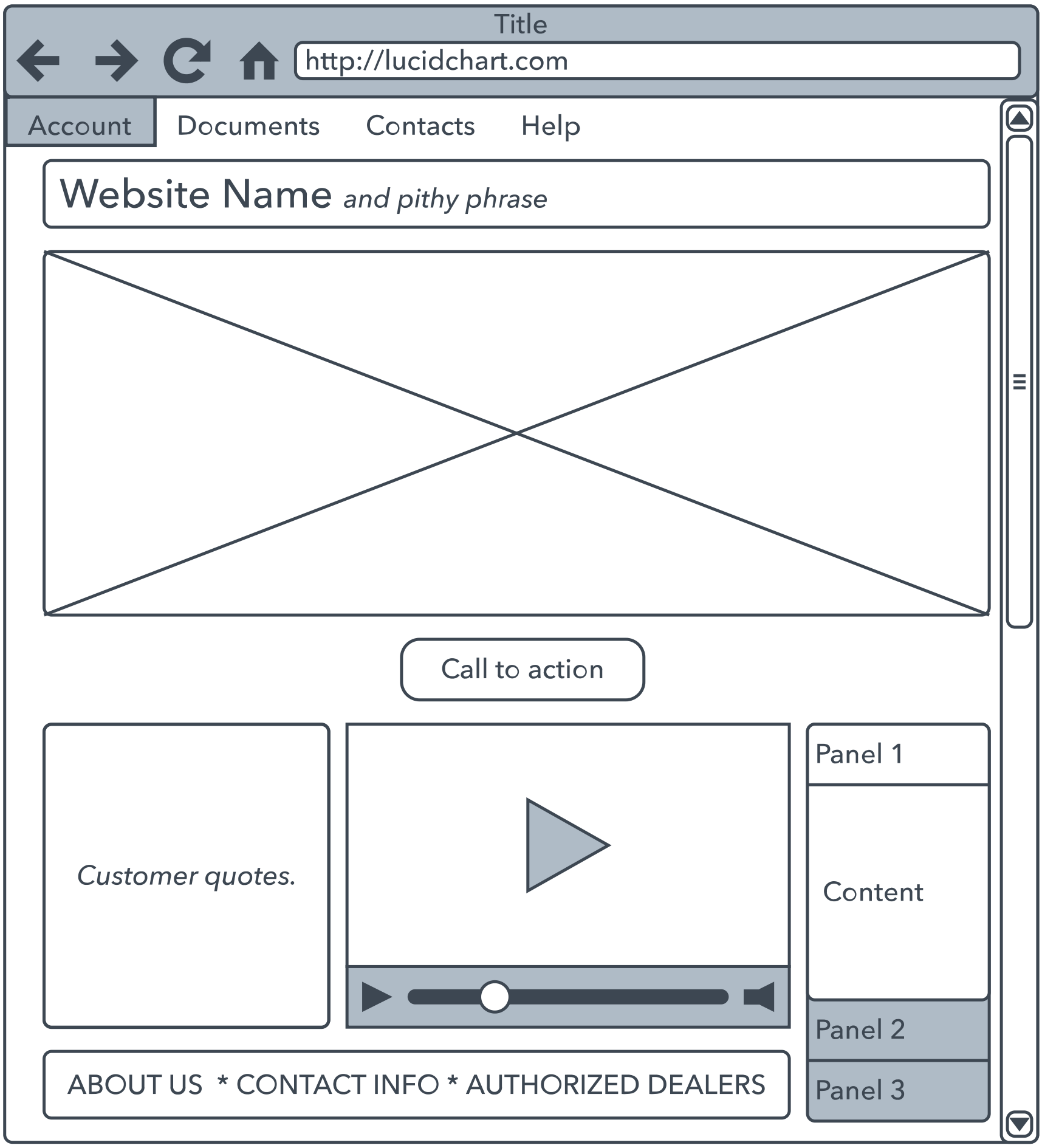


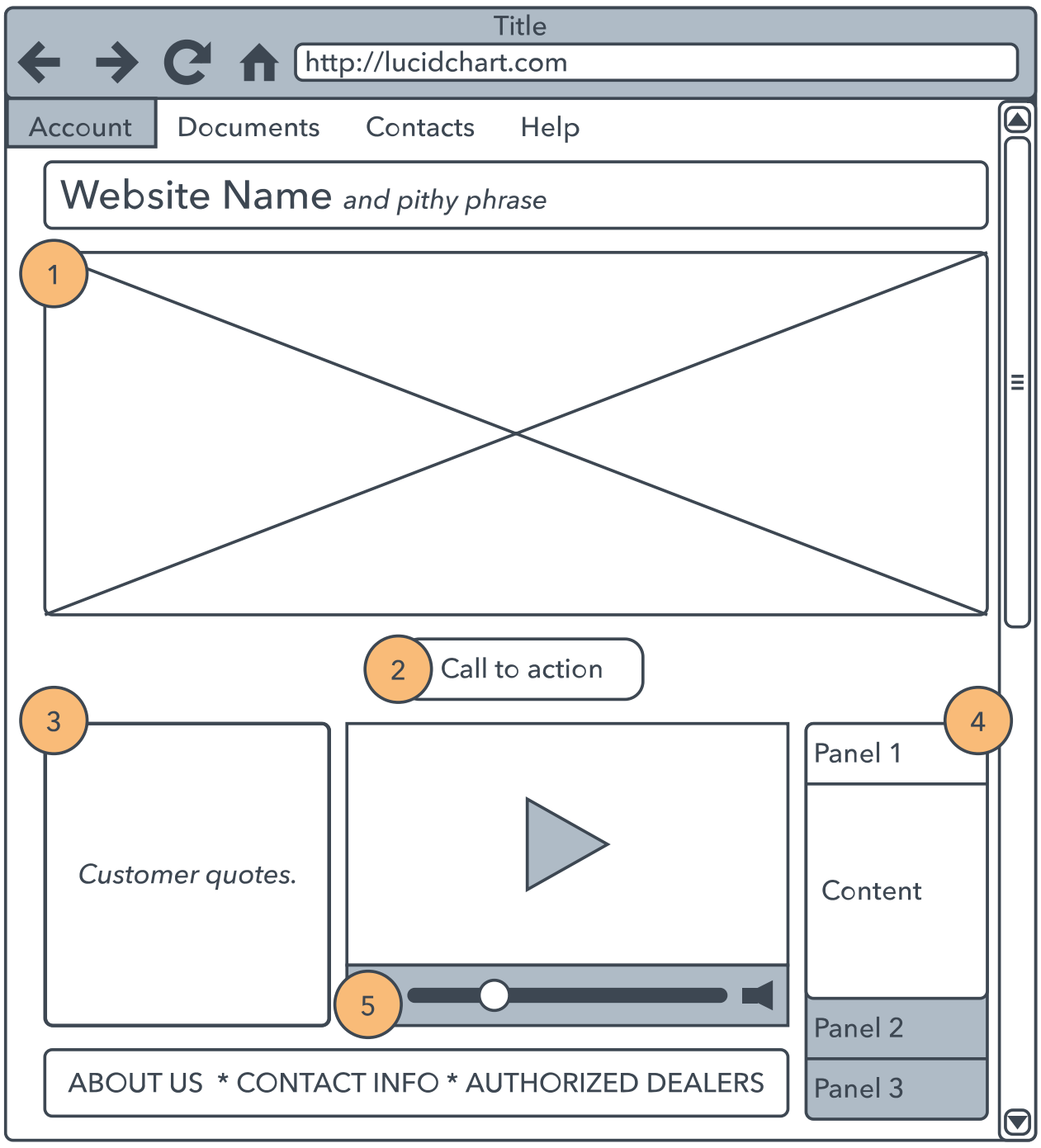
Source: https://en.wikipedia.org/wiki/Website_wireframe

Example Tools: LucidCharts



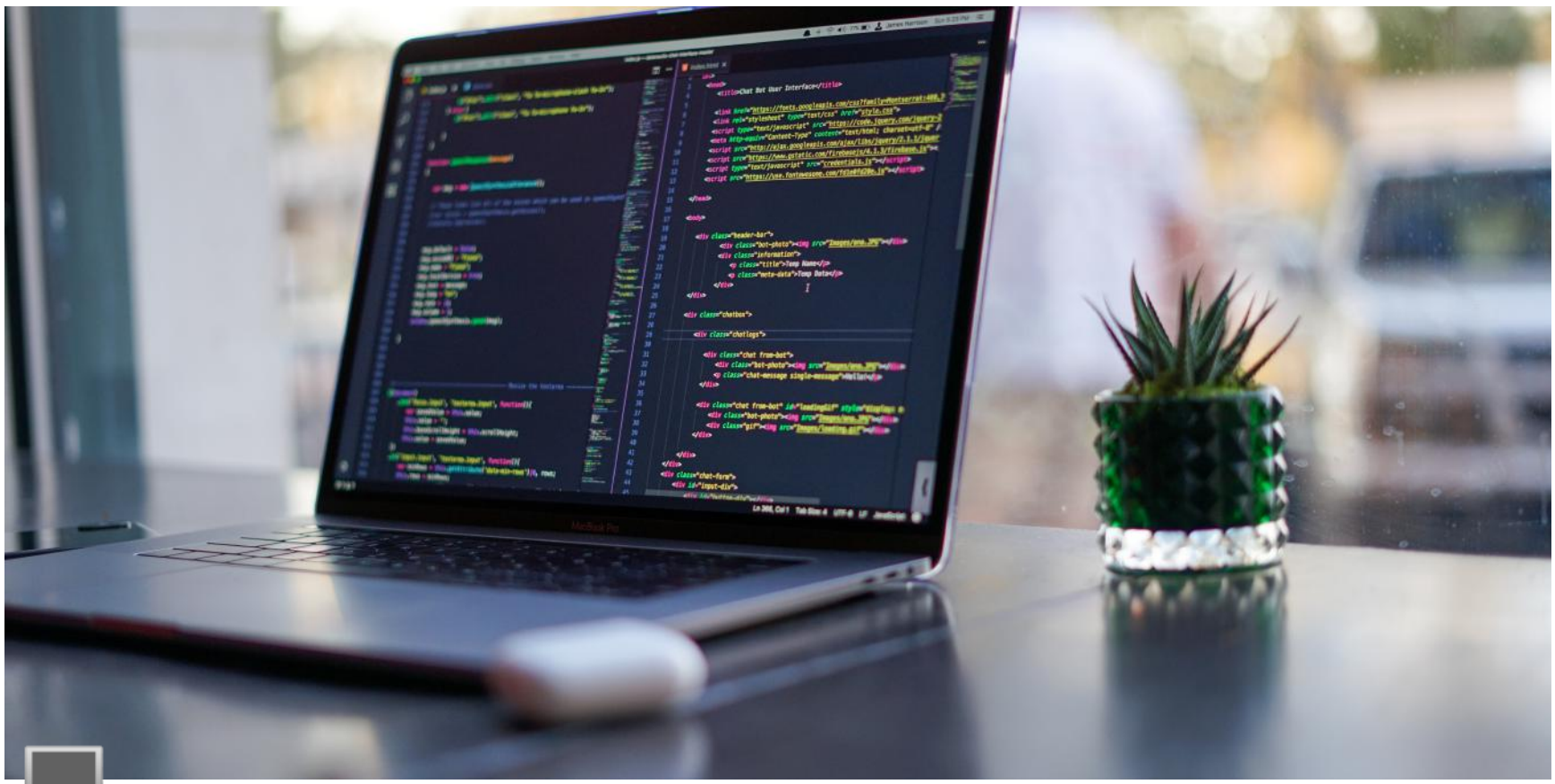
Source: https://www.lucidchart.com/pages/wireframe


Title
http://lucidchart.com
Account
Documents
Contacts
Help
Website Name and pithy phrase
Call to action
Customer quotes.
Panel 1
Content
Panel 2
Panel 3
ABOUT US * CONTACT INFO * AUTHORIZED DEALERS



Notes

1 Details about the image
2 Where this links to
3 List of possible quotes
4 Drop down menus that link to other pages
5 Content of video



Week 3 Lecture 6

Class	BSCCS2003
Created	@September 18, 2021 6:47 PM
Materials	
Module #	21
Type	Lecture
Week #	3

Tools (part 2)

Programmatic HTML generation: PyHTML

- Composable functions - each function generates a specific output
- Example:
  - To generate an `h1` heading: the function should return .. `"<h1>text of heading</h1>"`

```
import pyhtml as h

t = h.html(
  h.head(
    h.title('Test page')
  ),
  h.body(
    h.h1('This is a title'),
    h.div('This is some text'),
    h.div(h.h2('inside title'),
          h.p('some text in paragraph'))
  )
)
print(t.render())
```

More complex HTML

```
def f_table(ctx):
  return (
    tr(
      td(cell) for cell in row
```

```
    ) for row in ctx['table']
  )
```

Templates

- Standard template text
- Placeholder / Variables
- Basic (very limited) programmability
- Examples:
  - Python inbuilt string templates - good for simple tasks
  - Jinja2 - used by Flask
  - Genshi
  - Mako
  - ...

Jinja

- Ties in closely with Flask
- Template functionality with detailed API

  Remember: Templates can generate any output, not just HTML

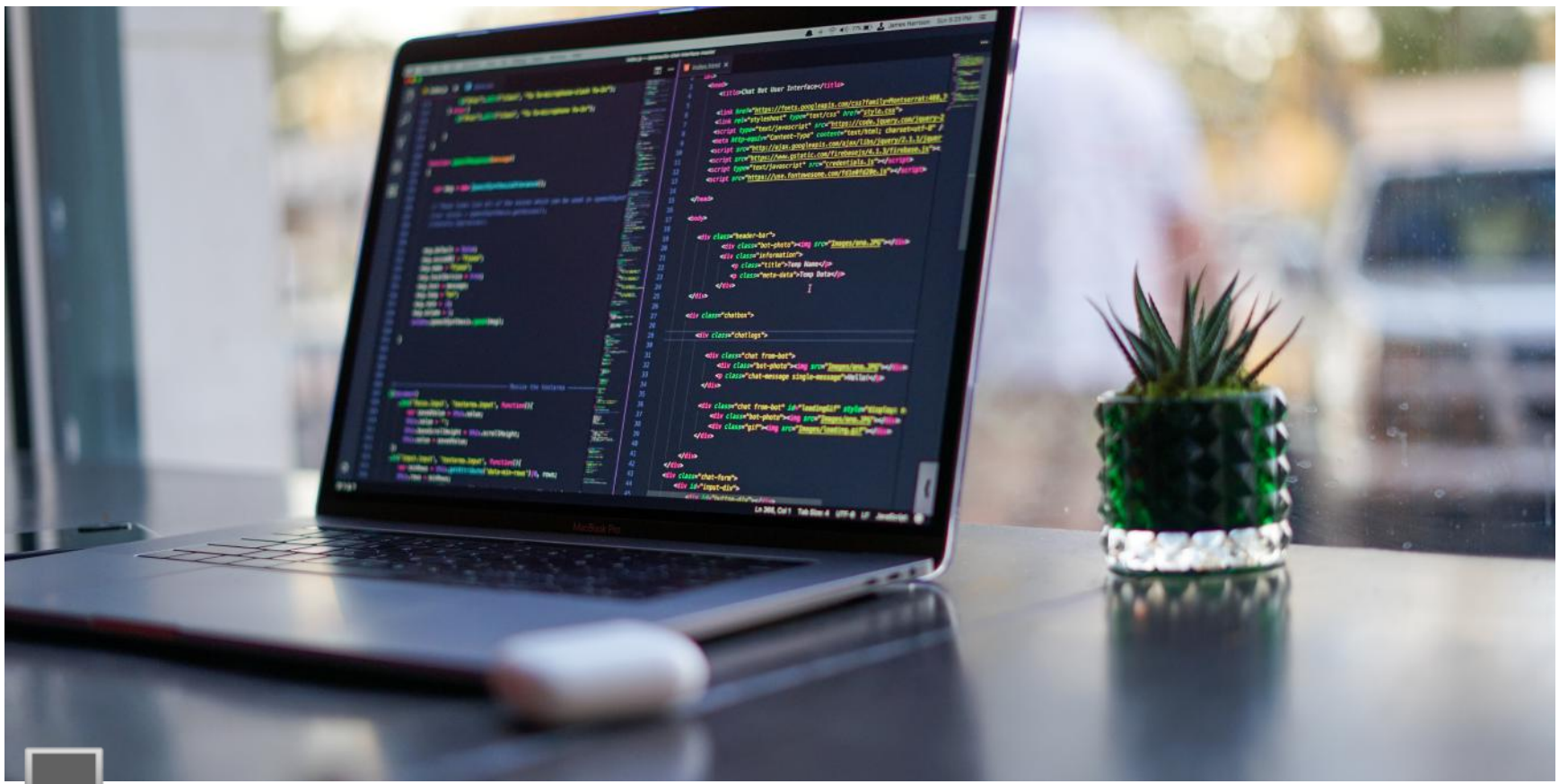
```
from string import Template

t = Template('$name is the $job of the $company')
s = t.substitute(name='Tim Cook', job='CEO', company='Apple Inc.')
print(s)
```

Jinja example

```
from jinja2 import Template
t = Template('Hello {{ something }}!')
print(t.render(something='world'))

t = Template('My favourite numbers: {% for n in range(1, 10) %}{{ n }} ' '{% endfor %}')
print(t.render())
```



Week 3 Lecture 7

Class	BSCCS2003
Created	@September 18, 2021 8:25 PM
Materials	
Module #	22
Type	Lecture
Week #	3

Accessibility

- Various forms of disability or impairment
  - Vision
  - Speech
  - Touch
  - Sensor-Motor
- Can a page be accessed by people with impairments?
- How can the accessibility of a page be improved?

W3.org - World Wide Web Consortium (W3C) - Accessibility guides

https://www.w3.org/WAI/fundamentals/accessibility-principles

Standards

Interplay between many components of a page:

- Web content: HTML, images, scripts, etc
- User-agents: Desktop browser, Mobile browser, speech-oriented browser, assistive devices
- Authoring tools: Text editor, Word processors, Compiler

Principle - Preceivable

- Provide text alternatives for non-text content
- Provide captions and other alternatives for multimedia

- Create content that can be presented in different ways, including by assistive technologies, without losing meaning.
- Make it easier for users to see and hear content

Principle - Operation

- Make all functionality available from a keyboard
- Give users enough time to read and use content
- Do not use content that causes seizures or physical reactions
- Help users navigate and find content
- Make it easier to use inputs other than keyboard

Principle - Understandable

- Make the text readable and understandable
- Make the content appear and operate in predictable ways
- Help users avoid and correct mistakes

Principle - Robust

- Maximize compatibility with current and future user tools

Aesthetics

- Visual appearance
- Very important
- Simplicity preferred

Can vary with time

Source:

iOS: A visual history

In what is widely regarded as his greatest presentation ever, Apple's Steve Jobs introduced the iPhone to the world on January 9th, 2007. In the five-plus years since then, the iPhone, iPad, and iPod Touch have literally redefined the entire world of mobile computing.

https://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad

Summary

- View - any screen seen by the human or machines
- User-interface and User-interaction guidelines
- Accessibility is a core concept
- Tools for automatic generation, consistent layout