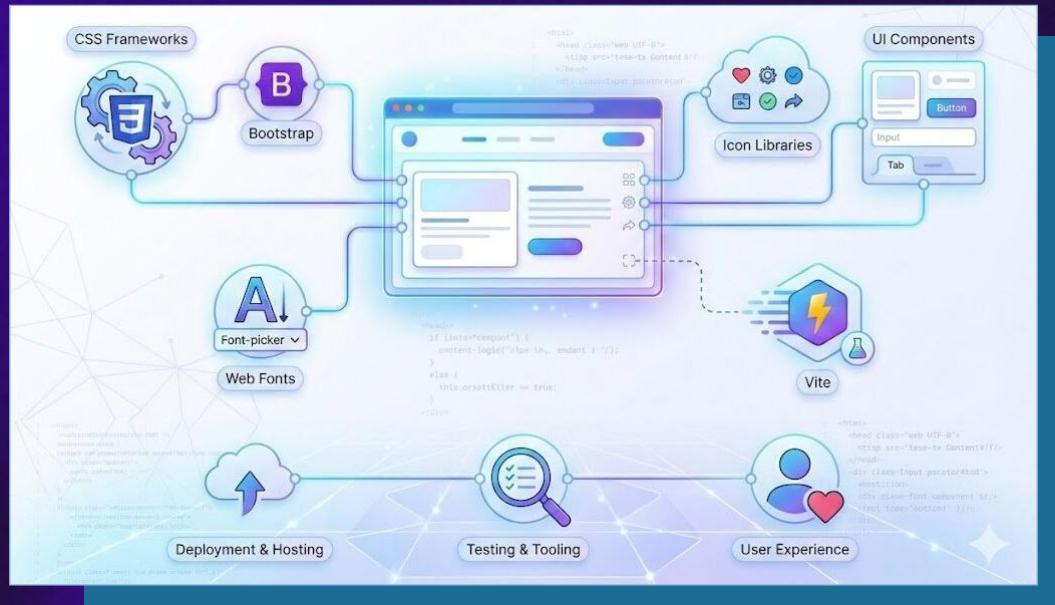


Web UI Technologies

CSS Frameworks, Bootstrap, Web Fonts, Icon Libraries, Vite, UI Components



Svetlin Nakov, PhD
Co-founder @ SoftUni

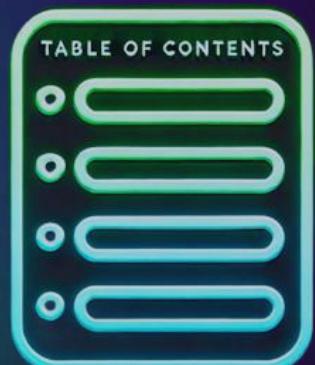


SoftUni AI

<https://ai.softuni.bg>

Agenda

- 1. CSS Frameworks:** Concepts, Bootstrap, Tailwind
- 2. Intro to Bootstrap:** Containers, Components, Layout
- 3. Bootstrap UI Components:** Navigation, Breadcrumbs, Tabs, Carousel, Accordion, Modal Popups, Drawer Sidebar, Tooltip, Toast Notifications, Loaders
- 4. Icons, Symbols, Effects:** Icons, SVG, Lucide Icons, Lotties
- 5. Web Fonts and Google Fonts Library**
- 6. Vite Dev Server:** Setup Vite Project, Multi-Page Apps
- 7. From Image to UI:** Screenshot / Figma to HTML page



Sli.do Code

#Soft-Tech-AI

Join at

slido.com

#Soft-Tech-AI



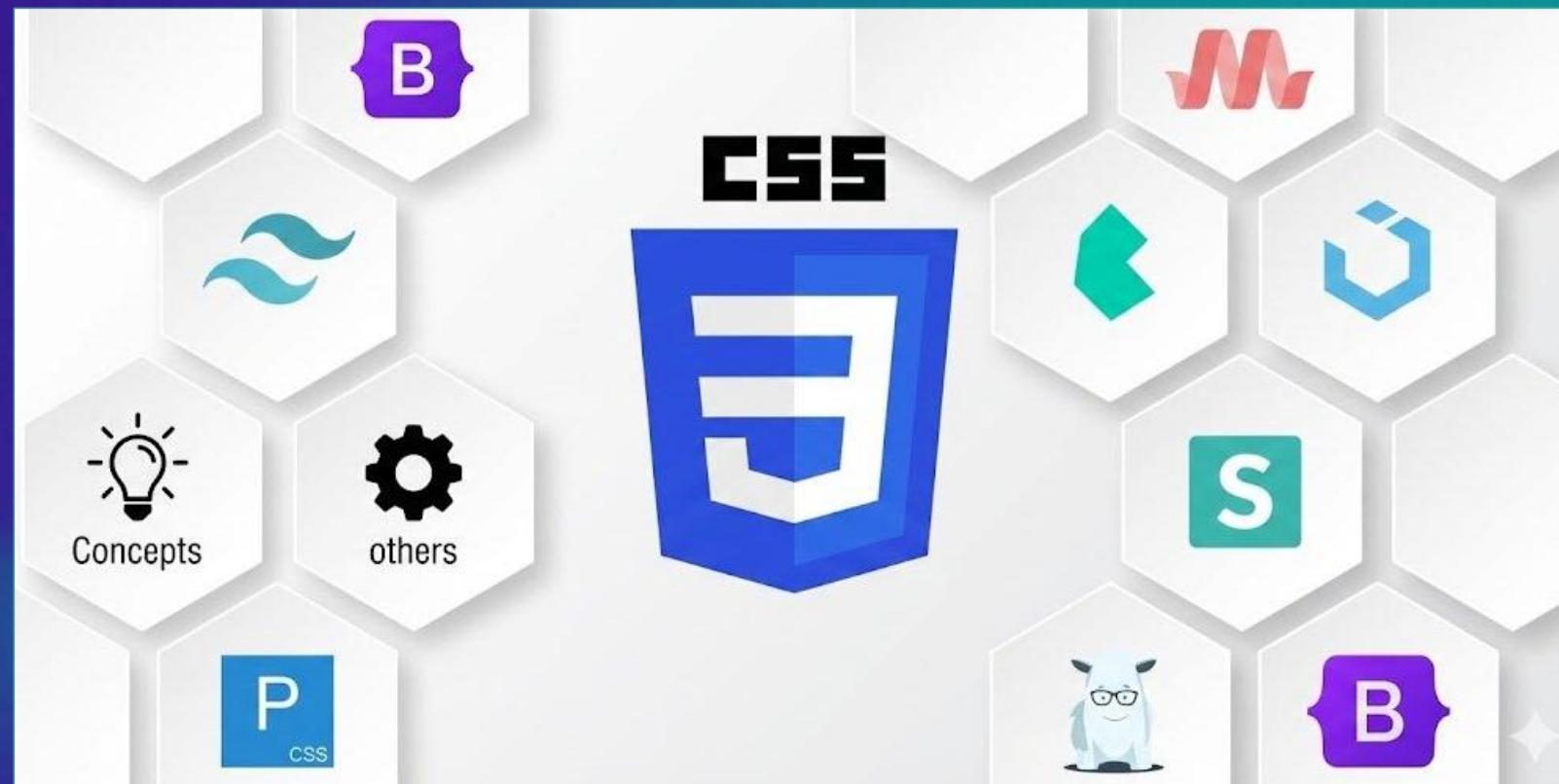
Breaks

20:00 / 21:00



CSS Frameworks

Concepts, Bootstrap, Tailwind



What is CSS Framework?

- **CSS framework** == prebuilt **library** of **styles**, **components**, and **utilities** to design consistent, responsive Web UIs
 - Instead of writing CSS from scratch, use **prebuild styles**
 - Use **prebuild library of components**: grid layout, hero section, navigation, tabs, accordion, drawer, etc.
- **How** to use a CSS framework?
 - Include prebuilt **CSS + JS** files to your HTML
 - Write less or zero custom CSS styles

CSS Frameworks: Examples

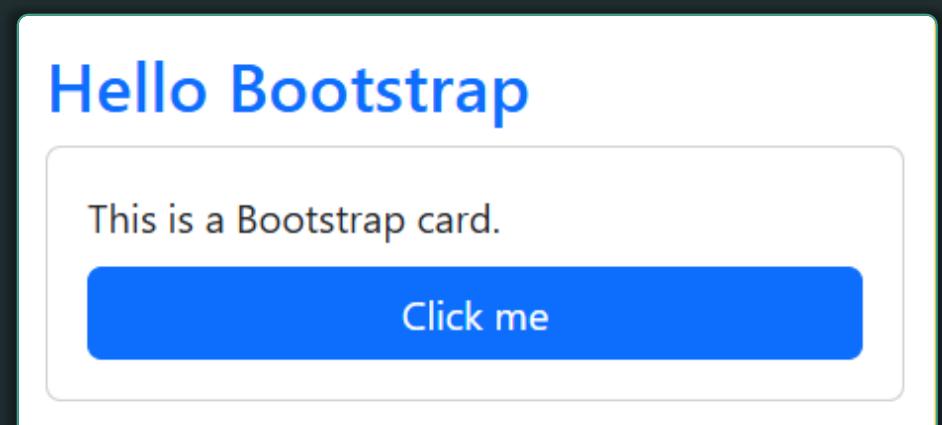
- Examples of CSS frameworks:
 - **Bootstrap**: the **most popular** CSS framework; component-based, responsive grid system, rich UI components
 - **Tailwind**: utility-first CSS framework → write *almost zero* custom CSS; ideal for **AI-generated UIs**
 - **Bulma**: modern, Flexbox-based CSS framework; class-only (no JavaScript), clean and readable syntax
 - **Materialize**: based on Google Material Design; ready-made components and animations

Bootstrap – Example

```
<link href= "https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

<div class="container mt-4">
  <h1 class="text-primary">
    Hello Bootstrap
  </h1>

  <div class="card p-3 mb-3">
    <p class="mb-2">This is a Bootstrap card.</p>
    <button class="btn btn-primary">Click me</button>
  </div>
</div>
```

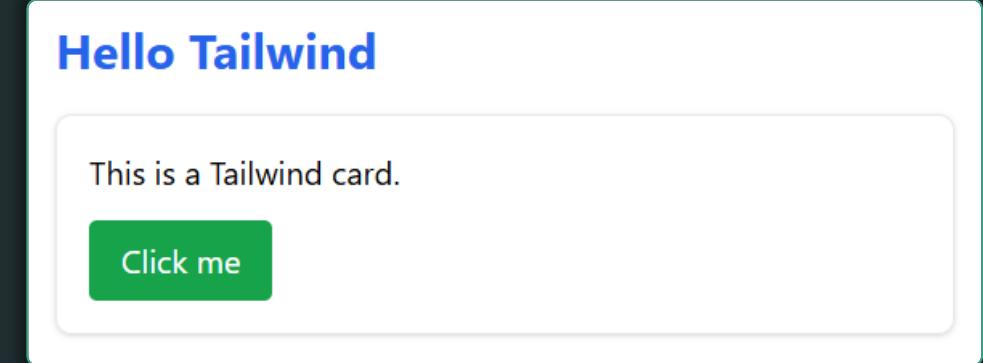


Tailwind – Example

```
<script src="https://cdn.tailwindcss.com"></script>

<div class="max-w-md mx-auto mt-8">
  <h1 class="text-blue-600 text-2xl font-bold mb-4">
    Hello Tailwind
  </h1>

  <div class="border rounded-lg p-4 shadow">
    <p class="mb-3">This is a Tailwind card.</p>
    <button class="bg-green-600 text-white px-4 py-2 rounded">
      Click me
    </button>
  </div>
</div>
```



Intro to Bootstrap

Bootstrap, Container, Components, Layout



The image shows a screenshot of the official Bootstrap website. At the top, there is a purple navigation bar with the Bootstrap logo (a white 'B' icon), followed by links to Home, Docs, Examples, Icons, Themes, and Blog. To the right of these are social media icons for GitHub, Twitter, Facebook, and Google+, and a 'Download' button. Below the navigation bar, the main content area has a white background. On the left, there is a large, bold, black text block that reads "Build fast, responsive sites with Bootstrap". To the right of this text is a graphic illustration featuring a large purple speech bubble containing a white 'B'. Behind the bubble are various icons representing mobile devices, a laptop, and other web development tools like a magnifying glass and a gear.

Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

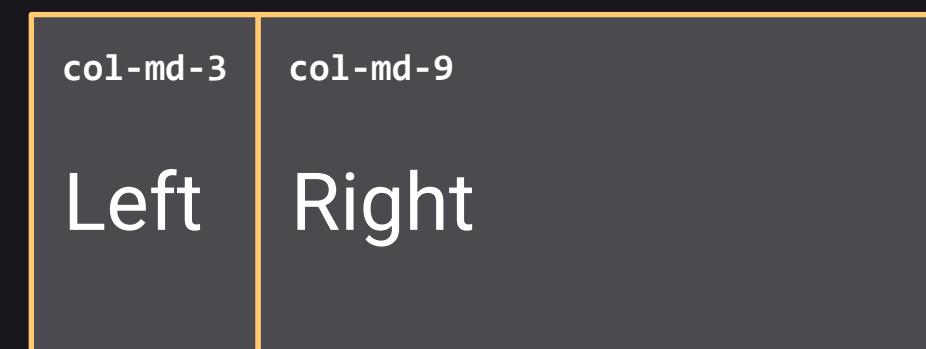
[Get started](#) [Download](#)

What is Bootstrap?

- **Bootstrap** == world's most popular CSS framework
 - Build **responsive, mobile-first websites** quickly
 - Ready-to-use **styles, layout tools**, and **UI components**
 - Powerful **12-column grid system** for layout



```
<div class="row">
  <div class="col-md-3">Left</div>
  <div class="col-md-9">Right</div>
</div>
```



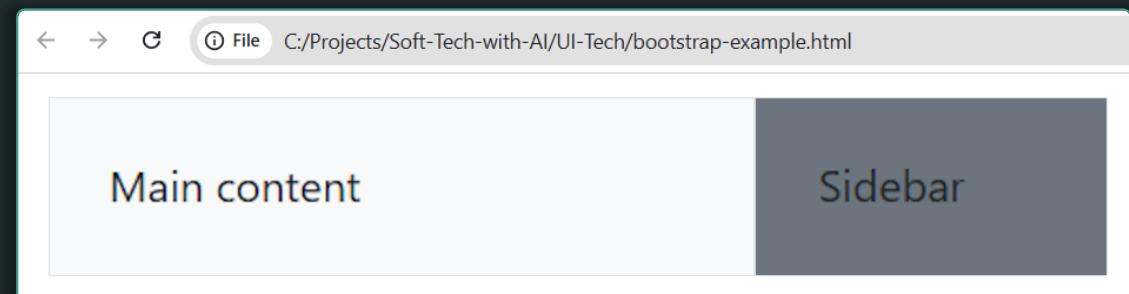
- Official site: <https://getbootstrap.com>

Getting Started with Bootstrap

- How to create a **Bootstrap-based Web page?**

```
<html><head>
  <link href= "https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/
  dist/css/bootstrap.min.css" rel="stylesheet">
</head>

<body>
  <div class="container mt-4">
    <div class="row">
      <div class="col-md-8 bg-light p-5 border">Main content</div>
      <div class="col-md-4 bg-secondary p-5 border">Sidebar</div>
    </div>
  </div>
</body></html>
```



Bootstrap Containers

```
<div class="container border bg-light p-3 mt-4">  
    Fixed-width container  
</div>
```

Fixed-width container

```
<div class="container-fluid border bg-light p-3 mt-4">  
    Full-width container  
</div>
```

Full-width container

```
<div class="container-md border bg-light p-3 mt-4">  
    Responsive container  
</div>
```

Responsive container

Responsive container

Bootstrap Components

```
<div class="col-md-8 bg-light p-5 border">
```

```
  <div class="card p-3 mb-3">
```

```
    <h5 class="card-title">Card title</h5>
```

```
    <p class="card-text">Some text.</p>
```

```
    <button class="btn btn-primary">Action</button>
```

```
  </div>
```

```
<div class="alert alert-info mb-3" role="alert">
```

```
  <strong>Info:</strong> This is a Bootstrap alert component.
```

```
  </div>
```

```
</div>
```

Card title

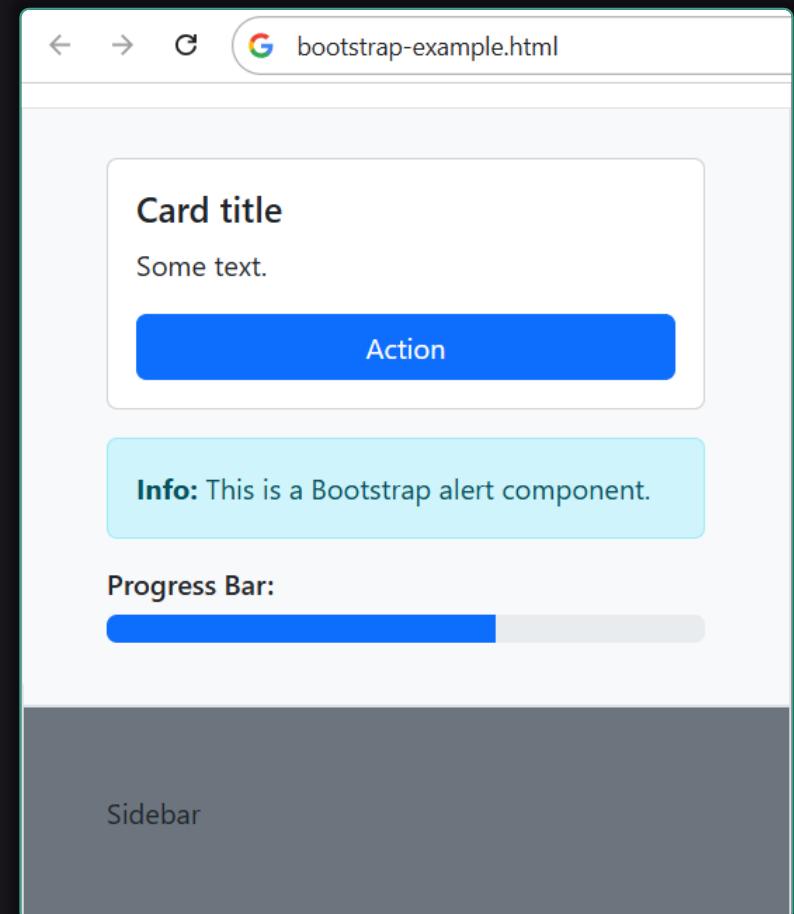
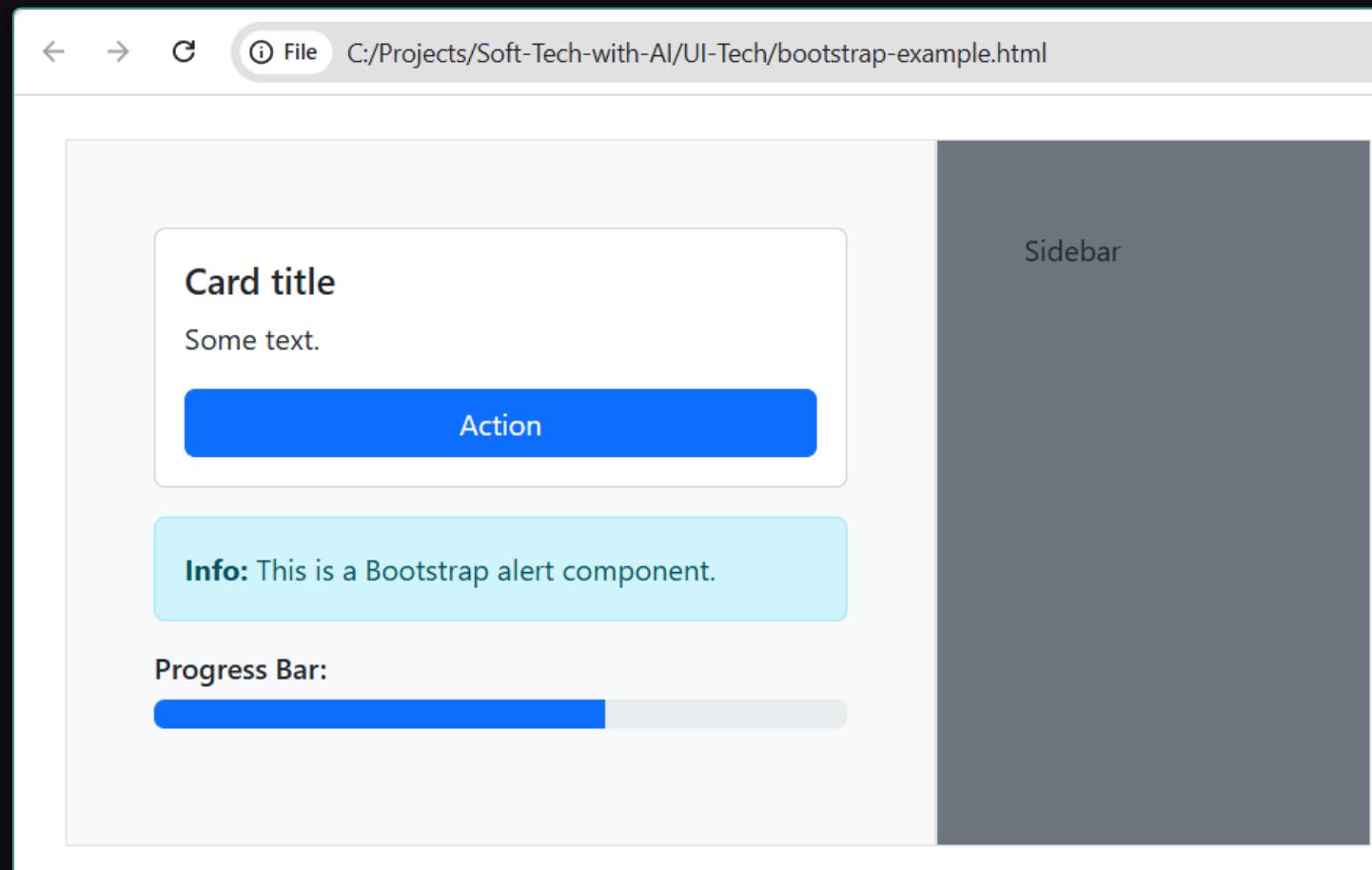
Some text.

Action

Info: This is a Bootstrap alert component.

Bootstrap and Responsive Design

- Bootstrap layouts are **responsive** by default:

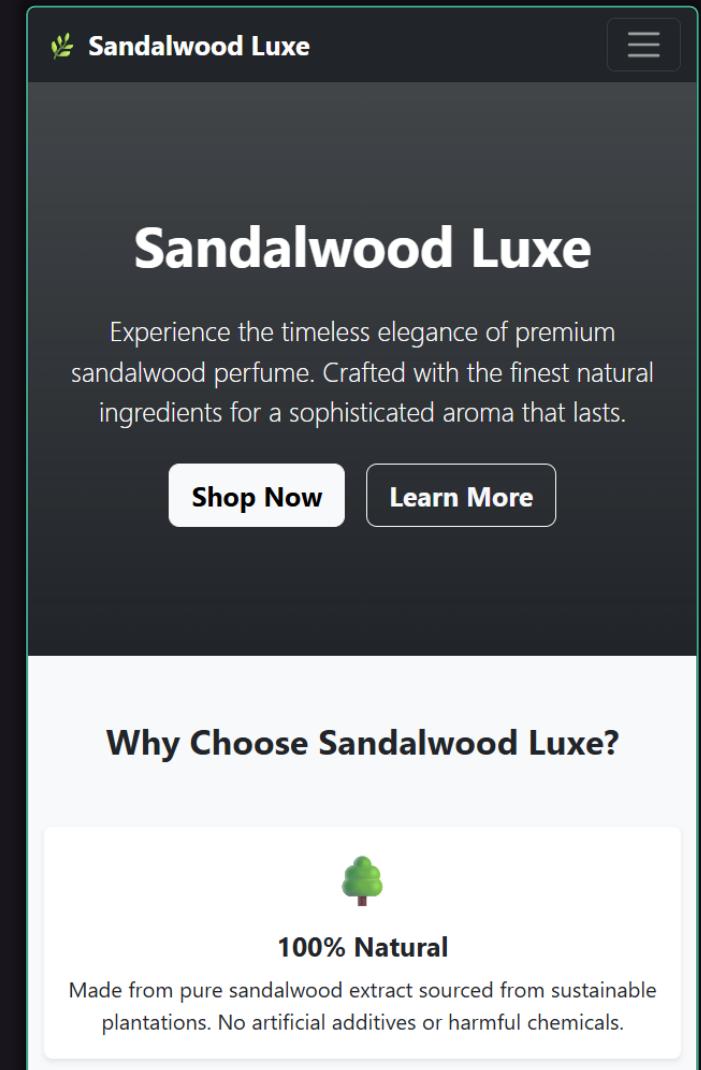


Exercise: Landing Page with Bootstrap

- Creating a **product landing page** with **Bootstrap**:

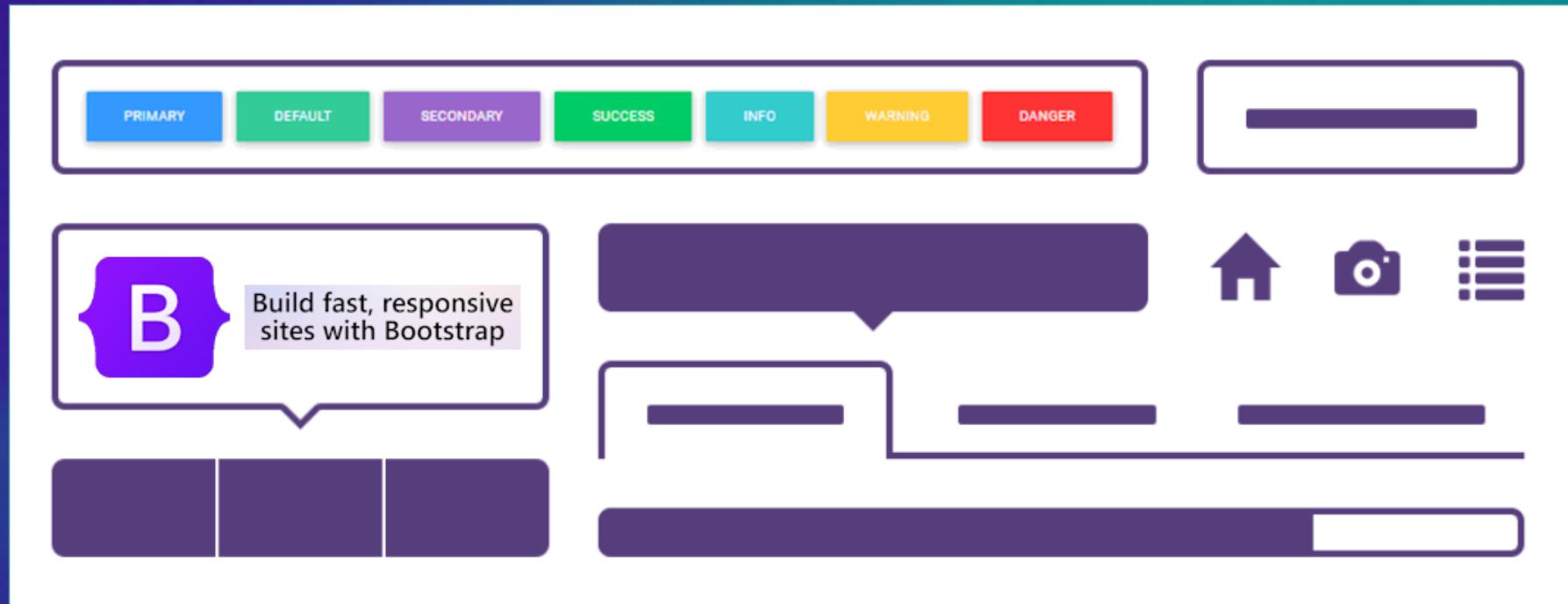
Create a **landing page** for certain product

- For example, a **perfume based on Sandal tree**
- Create an empty page: `perfume.html`
- Import **Bootstrap** via CDN
- AI generate the site **layout and navigation**
- AI generate a "**Hero**" section (banner with buttons)
- AI generate a "**Features**" section using **3 cards** in a row
- AI generate a "**Pricing**" table using Bootstrap styles
- Use only Bootstrap classes, no custom CSS



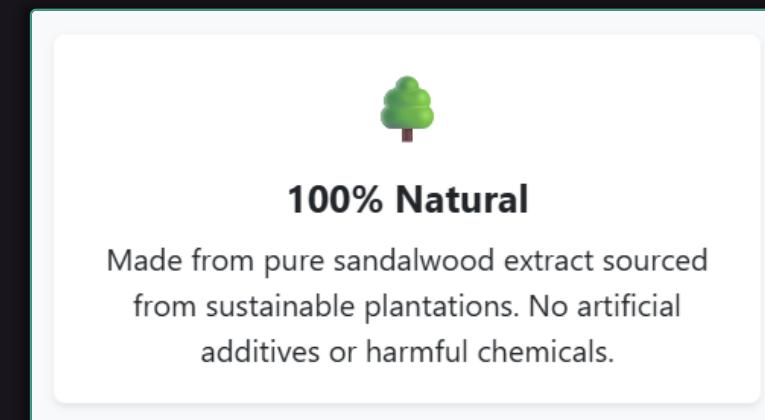
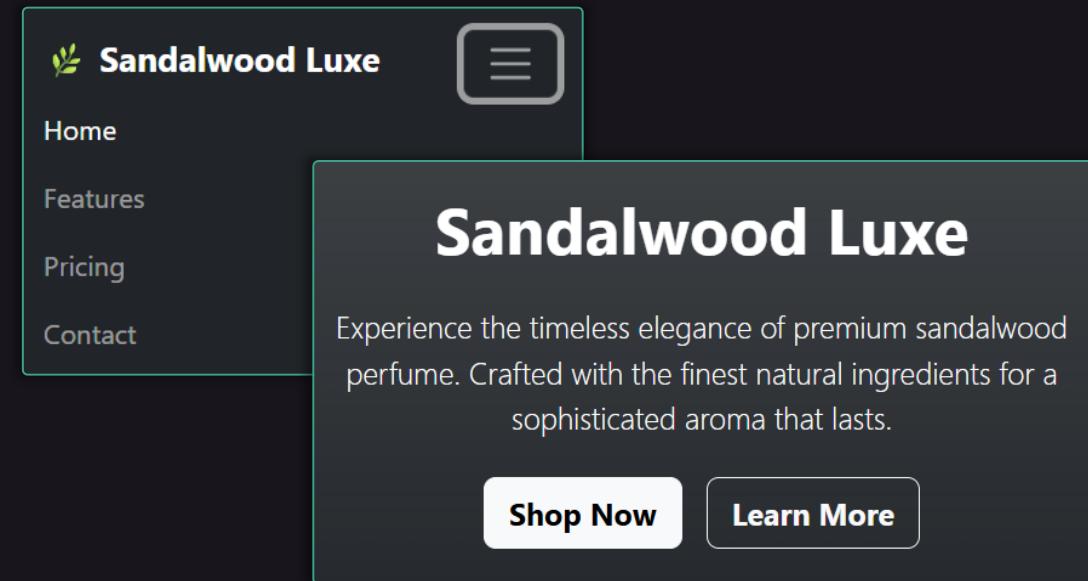
Bootstrap: Basic Components

Interactive Components, Navigation,
Breadcrumb, Tabs, Carousel, Accordion



Basic UI Components in Bootstrap

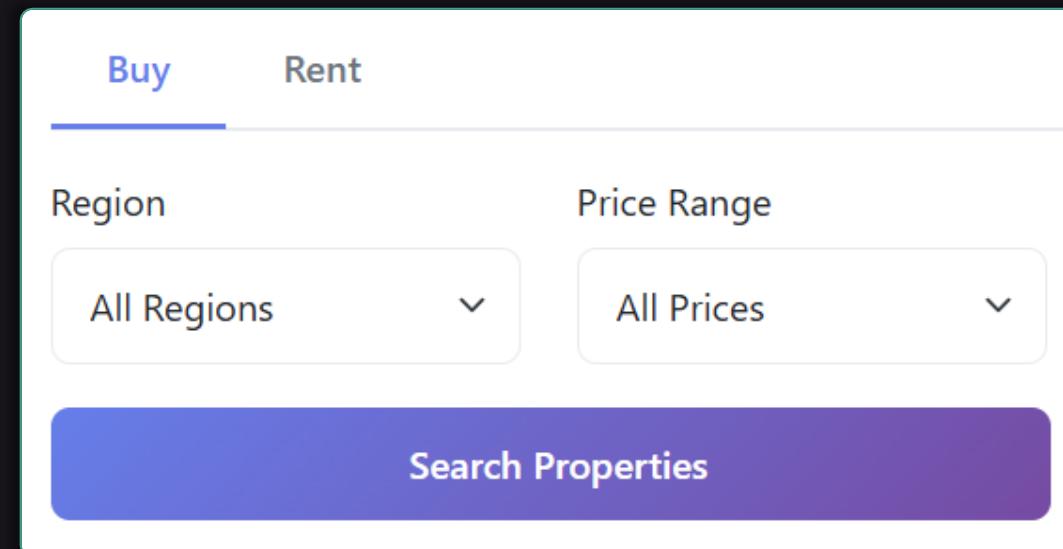
- Site Navigation: **Navbar** (responsive menu)
- **Hero Section** (headline + visual element + buttons)
 - The top banner of a site
- **Card** (box with shadow, image top, text below)
 - The fundamental building block of modern web



Basic UI Components in Bootstrap (2)

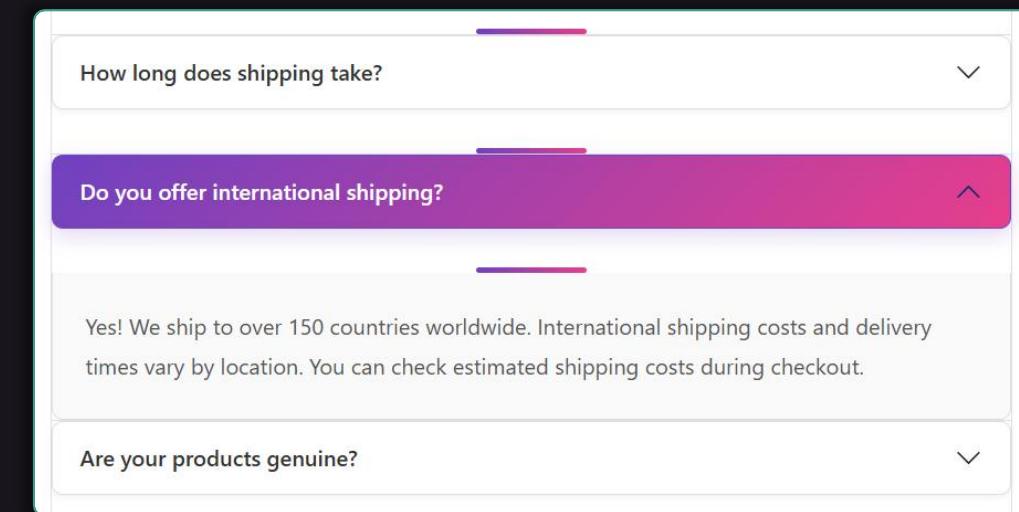
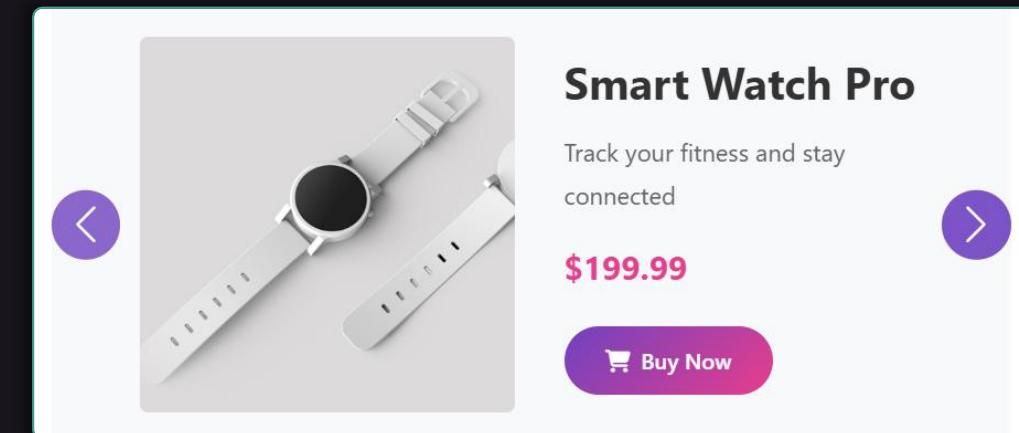
- **Breadcrumbs:** navigation trail
(e. g. Home > Properties > Find)
- **Tabs** (named boxes with switching between them)
 - Switch the view without reloading the page

Home / Properties / Find



Basic UI Components in Bootstrap (3)

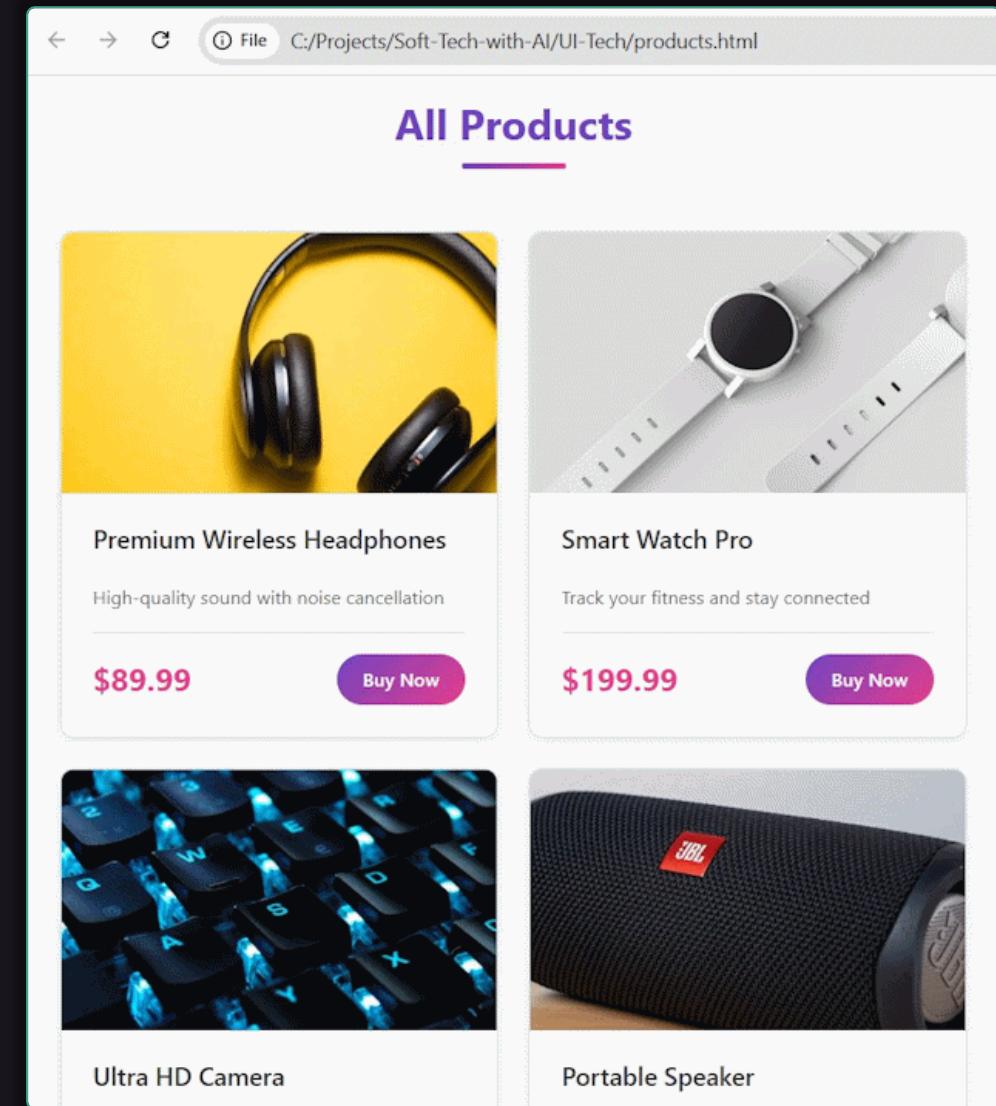
- **Carousel** (image slider)
 - Multiple cards / images with cycling through them
- **Accordion** (collapsible text)
 - A stack of items where clicking one expands the details (FAQ style)



Exercise: Products Page

Create a page `products.html` to display **products**

- Use HTML, CSS, JS and **Bootstrap**
- Create a **product card** component
 - Top image + title + price + [Buy Now] button
 - Add hover effects (button changes color, card slightly lifts-up)
- Implement a **standard layout**: header, navigation bar, main, footer
- Start with a "**Hero**" section
- Section "**Featured Productscarousel**
- Section "**All Productscards**
- Section "**FAQaccordion**
- Hold products in JavaScript array and render them dynamically at runtime

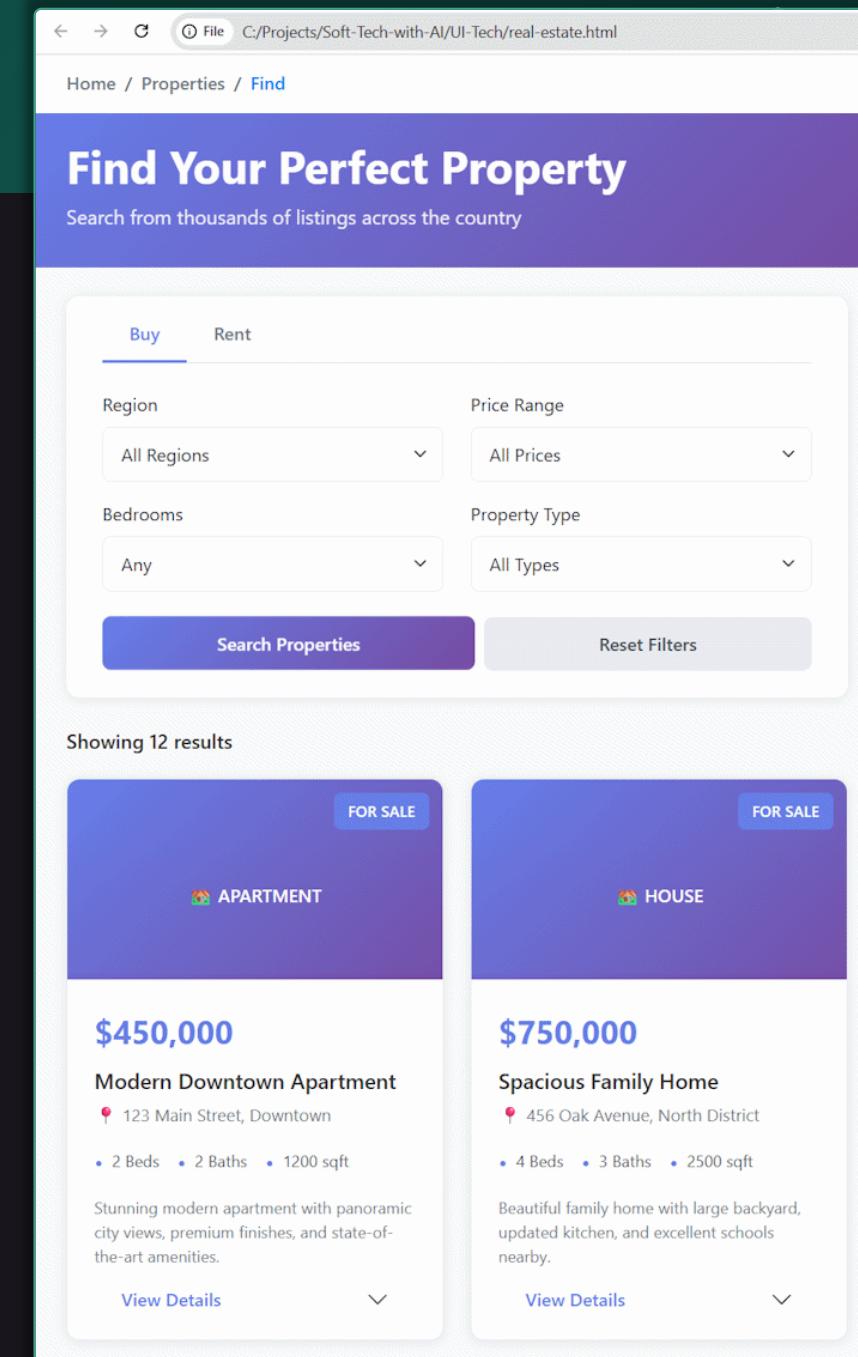


Exercise: Real Estate Page

Implement a simple Web page for a **real-estate** app:

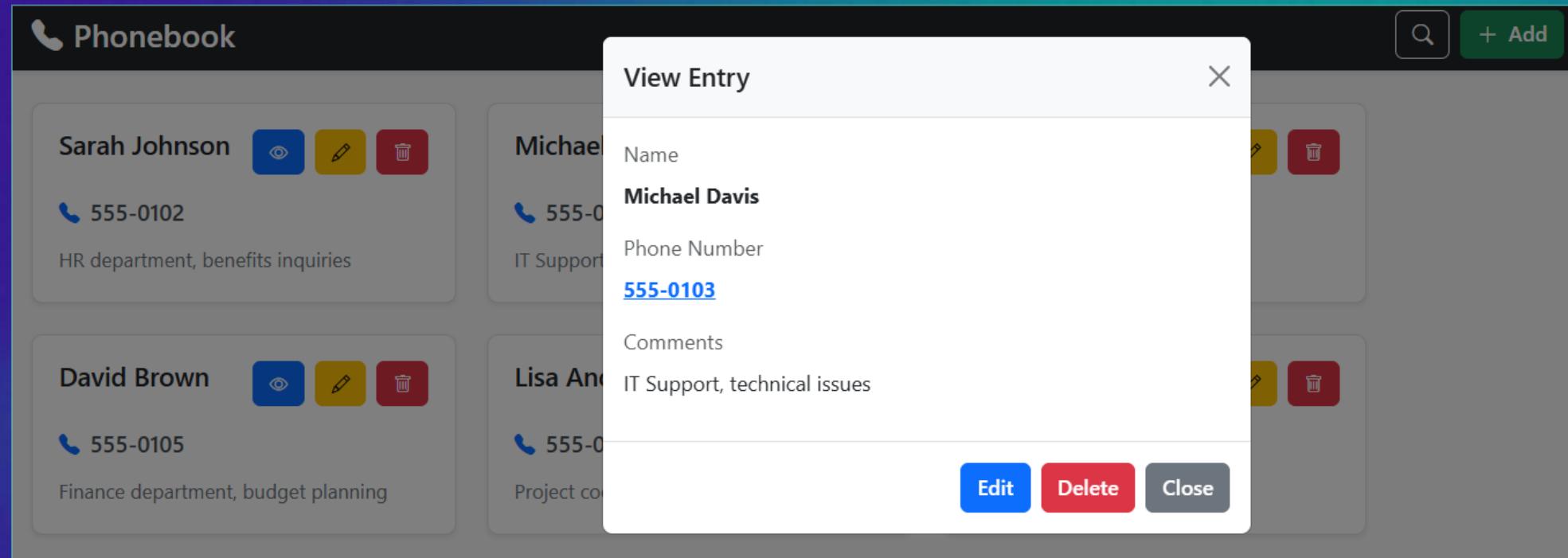
real-estate.html

- Use HTML, CSS, JS and **Bootstrap**
- Insert **breadcrumbs**: Home -> Properties -> Find
- The page starts with a "**Search**" section:
 - **Tabs**: Buy / Rent
 - Search form with **filters**: by region, price range, bedrooms, property type, etc.
- The next section is "**Search Results**":
 - Display the search results as **cards** + **paging**
 - Display property info in **accordion** "View Details"
- Hold estates in JavaScript array and render them dynamically at runtime



More Bootstrap Components

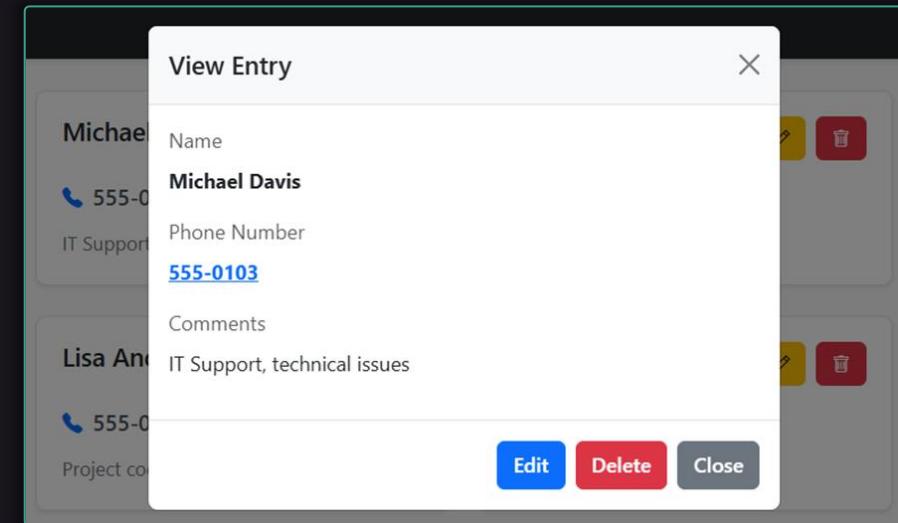
Modal Popups, Drawer Sidebar, Tooltip,
Toast Notifications, Loaders



Bootstrap UI Components

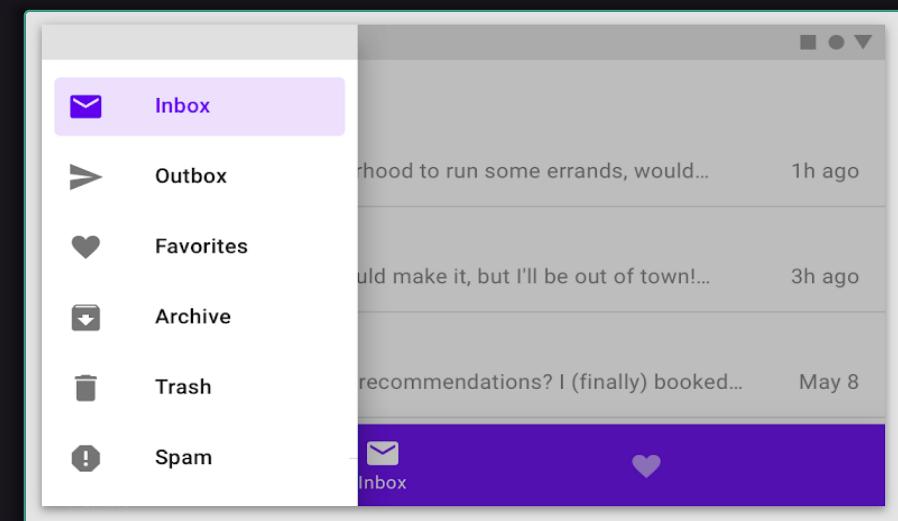
- **Modal Popup / Dialog:**

- A window that pops up and forces the user to interact
- Example: Confirm Delete → Yes / No
- The background usually goes dark (called a "backdrop")



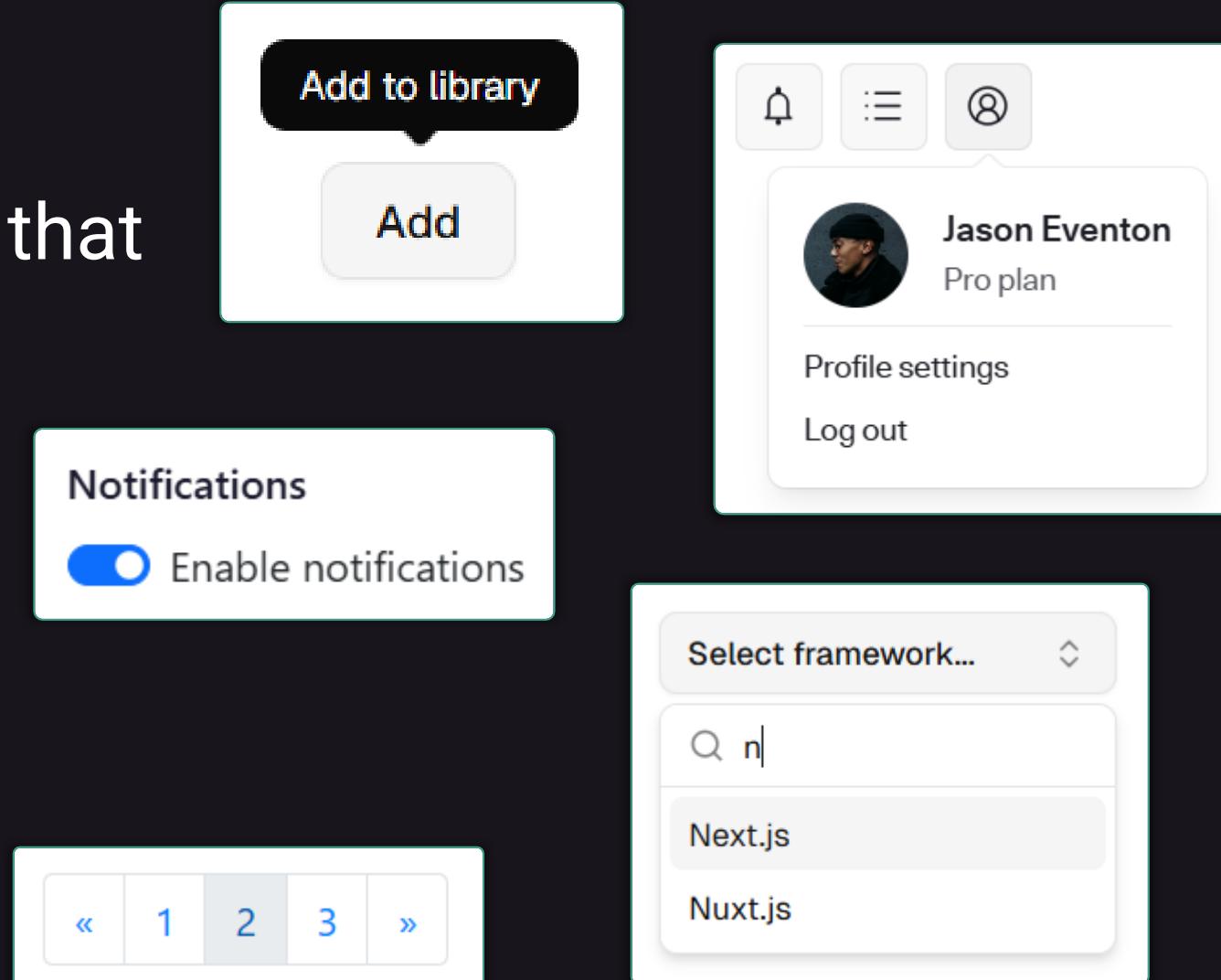
- **Drawer / Offcanvas / Sheet:**

- A panel that slides out from the left or right side of the screen
- Used for mobile menus or settings



Bootstrap UI Components

- **Tooltip**: a small hint that appears on icon hover
- **Popover**: clickable tooltip that shows interactive content
- **Switch (toggle button)**: modern On / Off slider
- **Combobox** (dropdown): a dropdown menu, optionally with search
- **Paginator**: paging UI

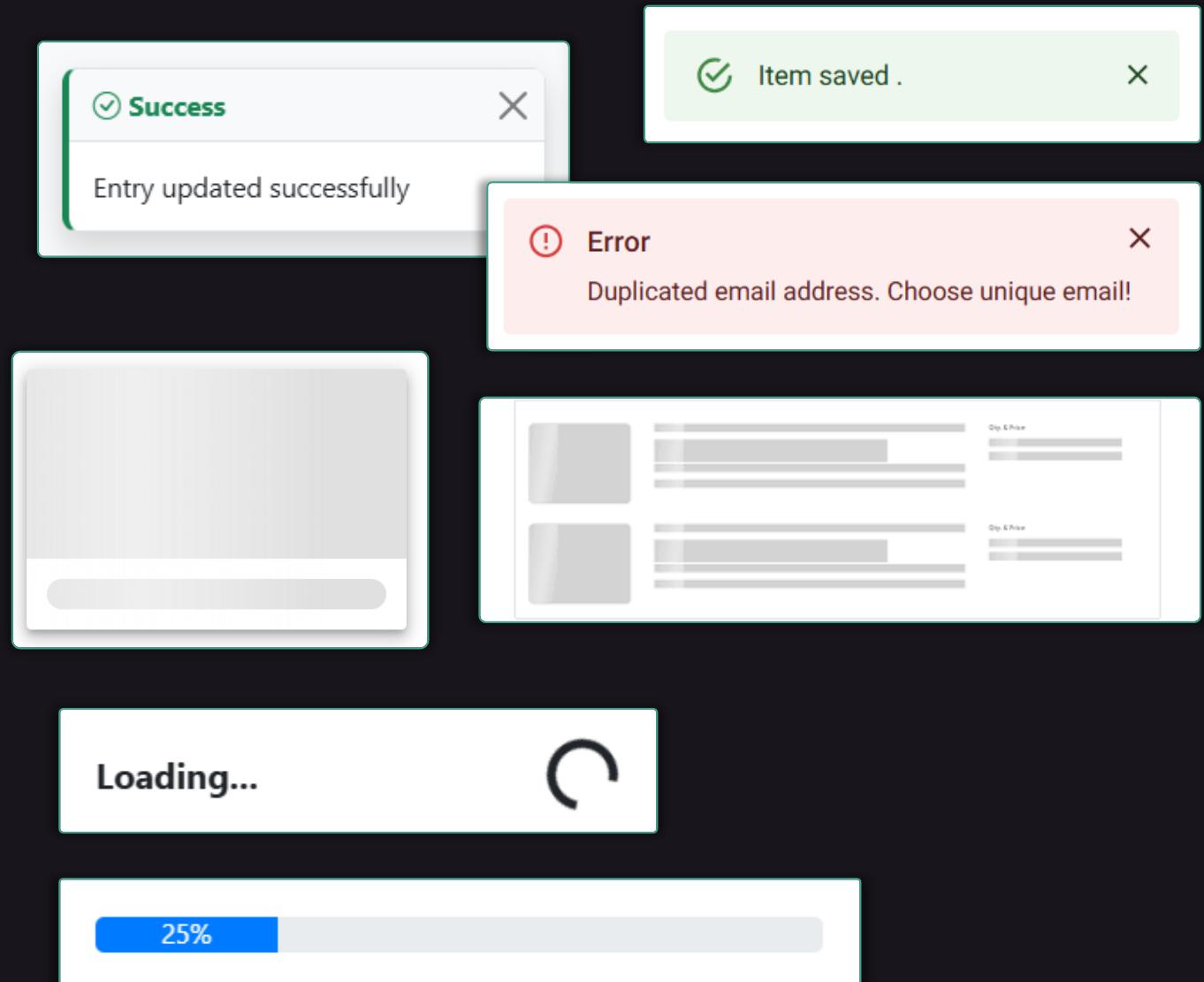


The image displays four examples of Bootstrap UI components:

- Tooltip:** A black speech bubble containing the text "Add to library" with a blue outline.
- Popover:** A larger, rounded rectangular box for Jason Eventon, showing profile settings and log out options.
- Switch:** A modern On/Off slider labeled "Notifications" with the text "Enable notifications".
- Combobox:** A dropdown menu titled "Select framework..." with search bar and items: "Next.js" and "Nuxt.js".
- Paginator:** A set of five numbered buttons with arrows for navigation: «, 1, 2, 3, ».

Bootstrap UI Components (2)

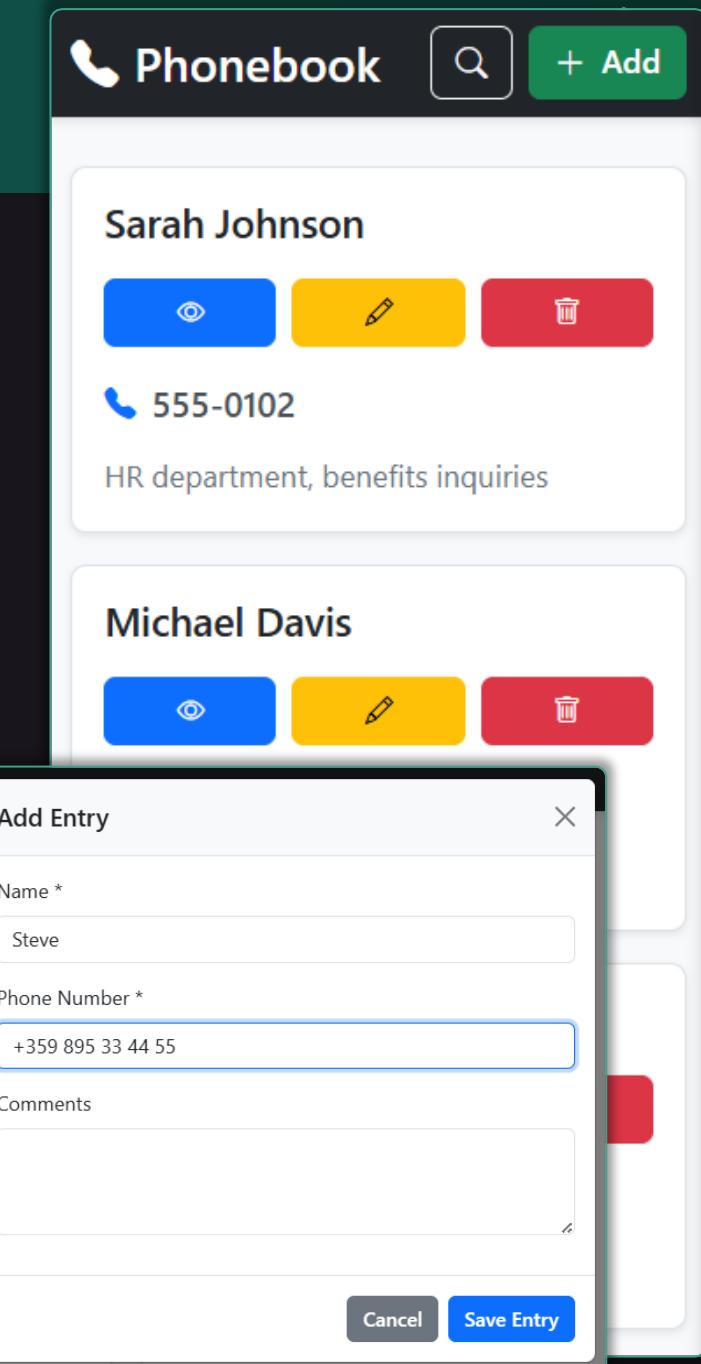
- **Toast Notification / Alert**
 - Info / warning / error message box
 - Auto dismiss / close
- **Placeholder / skeleton loader** / pulsing shape, shown during data loading
- **Spinner / loader**: rotating circle indicating "loading"
- **Progress bar** (with %)



Exercise: Phonebook App

Implement a **simple phonebook app** to manage phonebook entries (list, view, add, edit, delete).

- Use HTML + CSS + JS + **Bootstrap**
- Phonebook entry: name + phone number + comments
- Use separate files for the HTML, CSS and JS
 - **phonebook.html**, **phonebook.css**, **phonebook.js**
- Keep app data in the browser **local storage**
 - Initially generate 10 sample phonebook entries
- Implement list / view / add / edit of phonebook entries
 - Use **modal popup** for add / edit / delete confirmation
- Action **buttons**: implement as **icon + tooltip**
- Use **toast notifications** after each operation



Interactive Dashboard with Bootstrap

Create a simple **admin dashboard view**

- Create a page **dashboard.html** + **dashboard.css**

- Import **Bootstrap** via CDN

- Implement **sidebar navigation**

with **navigation buttons**

- Collapsible sidebar

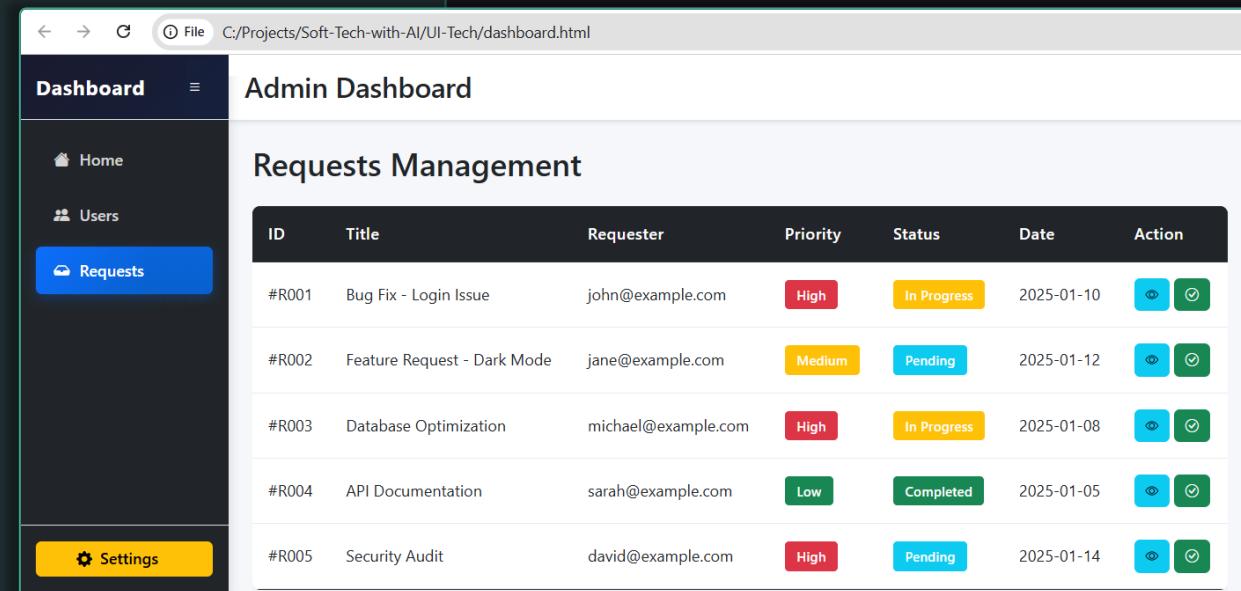
- App **tabs** hold different content:

- **Home tab**, holding statistics

- **Users tab**, holding a data table

- **Requests tab**, holding a data table

- A "**Settings**" button that opens a **modal popup**



The screenshot shows a web browser displaying the 'Admin Dashboard' at the URL 'C:/Projects/Soft-Tech-with-AI/UI-Tech/dashboard.html'. On the left, a dark sidebar titled 'Dashboard' contains three tabs: 'Home', 'Users', and 'Requests', with 'Requests' being the active tab. At the bottom of the sidebar is a yellow 'Settings' button. The main content area is titled 'Admin Dashboard' and 'Requests Management'. It features a data table with columns: ID, Title, Requester, Priority, Status, Date, and Action. The table contains five rows of data:

ID	Title	Requester	Priority	Status	Date	Action
#R001	Bug Fix - Login Issue	john@example.com	High	In Progress	2025-01-10	 
#R002	Feature Request - Dark Mode	jane@example.com	Medium	Pending	2025-01-12	 
#R003	Database Optimization	michael@example.com	High	In Progress	2025-01-08	 
#R004	API Documentation	sarah@example.com	Low	Completed	2025-01-05	 
#R005	Security Audit	david@example.com	High	Pending	2025-01-14	 

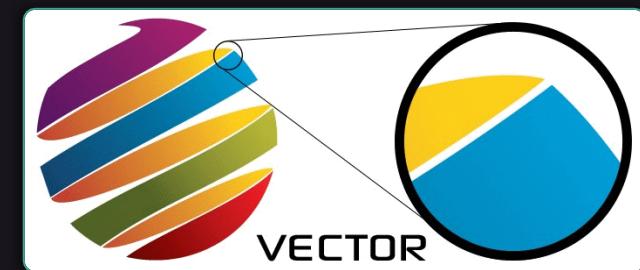
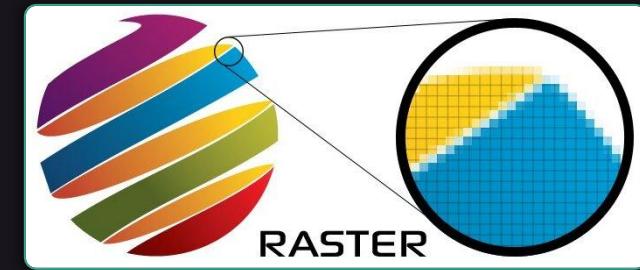
Icons, Symbols, Effects

Icons, Symbols, SVG Graphics, Lucide
Icons, Lotties, Transitions and Effects



Icons, Symbols and SVG

- **Icons** are small images, used in the app UI elements (buttons, toolbars, form controls)
- **Images** are raster objects (consist of **pixels**)
 - Lose quality when zoom-in (pixelization)
 - File formats: **PNG, JPG, GIF, WebP, AVIF**
- **Icons** and **SVG graphics** are vector objects
 - Can **zoom-in infinitely** without quality loss
 - File formats: **SVG, PDF, EPS, AI**
 - Can hold **symbols, icons, diagrams, company logos**



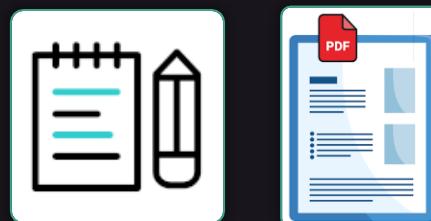
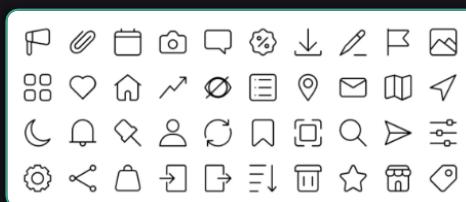
SVG Graphics – Example

```
<h1>SVG Example</h1>

<svg width="100" height="100"
viewBox="0 0 24 24" fill="none"
stroke="#8f3be4" stroke-width="2"
stroke-linecap="round" stroke-
linejoin="round">
  <path d="M3 9l9-7 9 7v11a2 2 0
0 1-2 2H5a2 2 0 0 1-2-2z"></path>
  <polyline points="9 22 9 12 15
12 15 22"></polyline>
</svg>
```



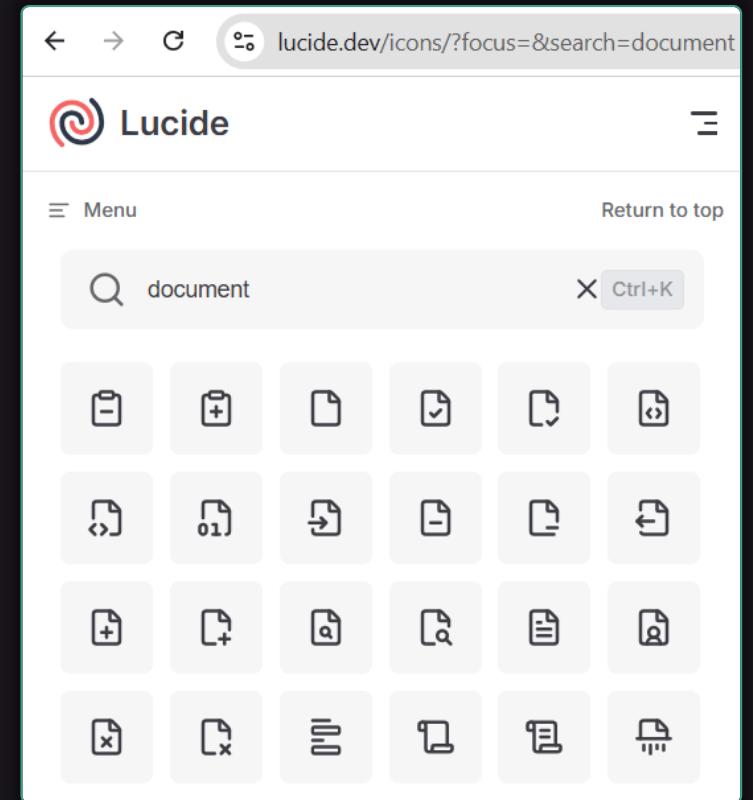
Images, Icons, SVG, Emojis, PDF



- **Graphics** in modern Web:
 - **Images** for photos, e. g. a photo of product, person
 - **SVG** for graphical symbols like company logo, UI graphics, technical and functional symbols, diagrams
 - **Emojis** (Unicode chars) for simple icons in the text
 - **Icon libraries** for more complex icons
 - **Lotties** (animated icons) for visual communication
 - **PDF** for documents (brochures, books, reports, ...)

Icon Libraries

- **Icon library** – a large set of **vector icons** for Web
 - Examples: **Lucide Icons**, **Font Awesome**, **Heroicons**, **Material Design Icons**, **Iconify**
- **Lucide Icons**
 - 2000+ free icons in 50+ categories
 - <https://lucide.dev>
 - JS packages for Vanilla JS, React, Angular, Vue, React Native, Svelte, ...



Lucide Icons – Example

```
<html>
<head>
  <title>Lucide Icons</title>
  <script src="https://unpkg.com/lucide@latest"></script>
  <style>
    .icon-item {
      display: flex;
      align-items: center;
      gap: 0.3rem;
    }
  </style>
</head>
```

```
<body>
  <h1>Lucide Icons</h1>
  <div class="icon-item">
    <i data-lucide="home"></i> Home
  </div>
  <div class="icon-item">
    <i data-lucide="search"></i> Search
  </div>
  <div class="icon-item">
    <i data-lucide="settings"></i> Settings
  </div>
  <script>lucide.createIcons();</script>
</body>
</html>
```

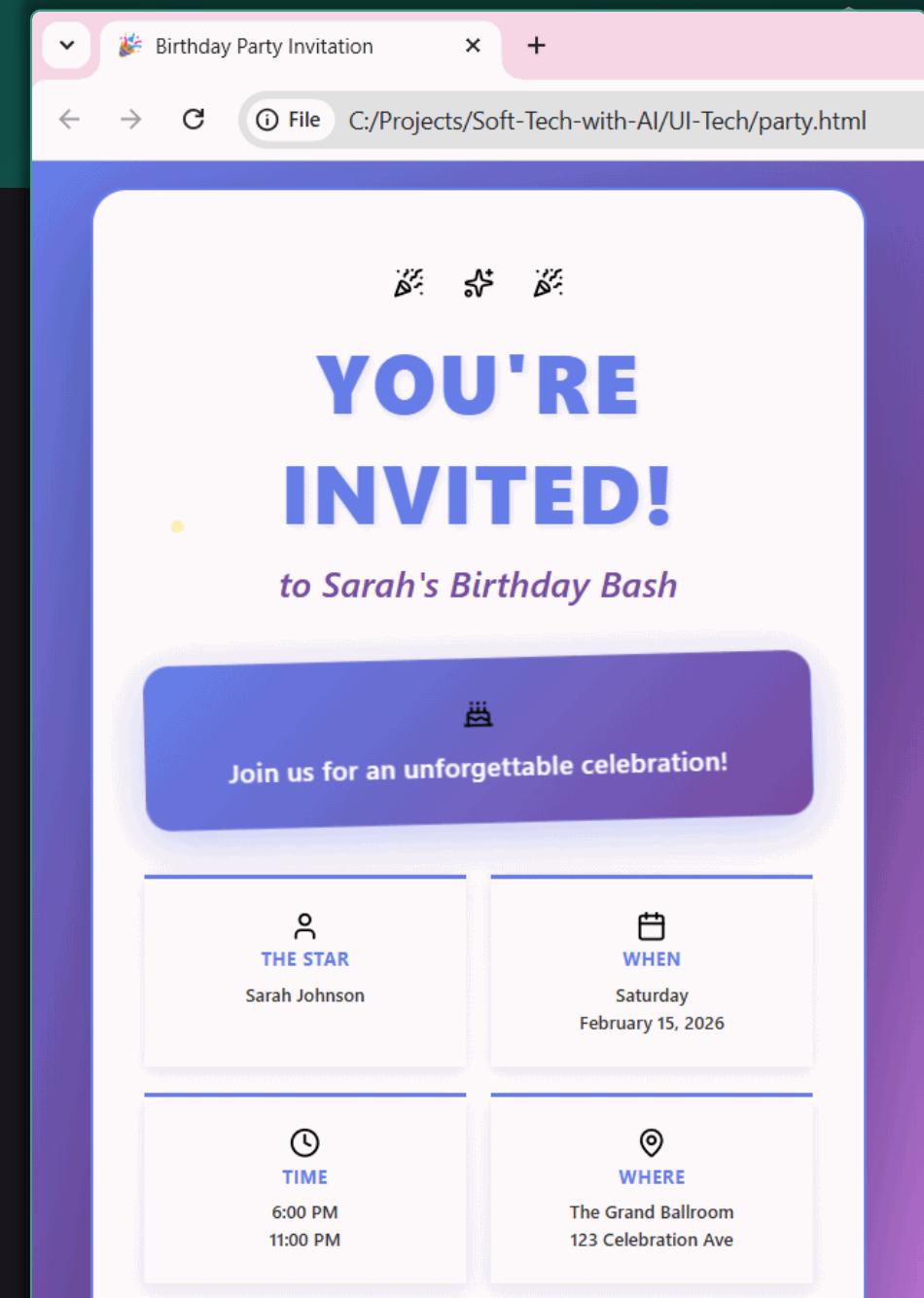
Lucide Icons

-  Home
-  Search
-  Settings

Exercise: Party Page

Party Invitation

- Create a modern birthday party invitation page
 - `party.html` + `party.css`
 - Design the invitation like a modern poster
 - Use HTML, CSS and **Lucide Icons**
 - Use **icons** to enhance the visual communication
 - Insert "favicon" icon for the Web page



Lotties

- **Animated icons** and graphics can improve the visual communication of your Web site or app
- **Animated GIFs** – heavy (large file size), raster graphics
- **Videos** – heavy (very large file size), slow to load & play
- **Lotties** – lightweight vector animations, load very fast
 - Created with Adobe After Effects / Figma / LottieFiles
 - Saved as **JSON files**, played with **Lottie JS library**
 - Lottie libraries, e. g. <https://lottiefiles.com>

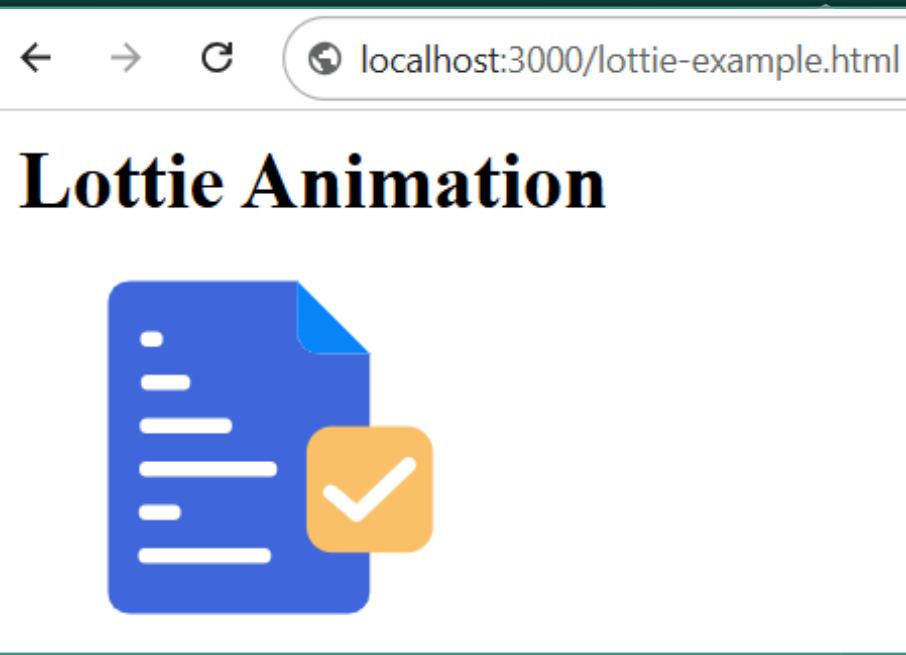


Lottie – Example

```
<!DOCTYPE html>
<html>
<body>
  <h1>Lottie Demo</h1>
  <script src=
    "https://unpkg.com/@dotlottie/player-component"></script>
  <dotlottie-player src="document-lottie.json" autoplay loop
style="width: 200px;" />
</body>
</html>
```

Terminal: npx serve

```
document-lottie.json
lottie-animation > {} document-lottie.json > ...
{"v":"4.8.0","meta": {"g":"LottieFiles AE 3.0.2","a":"","k":"","d":"","tc":""}, "fr":30, "ip":0, "op":101, "w":500, "h":500, "nm":"Document", "ddd":0, "assets":[], "layers":[{"ddd":0, "ind":1, "ty":4, "nm":"check_mark", "sr":1, "ks": {"o": [{"a":0, "k":100, "ix":11}, {"r":{"a":0, "k":0, "ix":10}, "p":{"a":0, "k": [348.45, 291.732, 0]}, "ix":2}, {"a":0, "k": [98.45, 41.732, 0]}, {"ix":1}, {"s": [{"a":0, "k": [100, 100, 100]}, {"ix":6}]}]}]}
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS UI-Tech> npx serve
Serving!
- Local: http://localhost:3000
- Network: http://10.5.0.2:3000
Copied local address to clipboard!
```

Transitions and Effects

- **Transition** – a smooth change between two states
 - Example: button lifts and changes its color on hover
- **Effect** – visual enhancements to guide attention and communicate interactivity
 - Example: elements slide in as you scroll down
- **Microinteractions** – enhance visual feedback
 - Example: success checkmark when you fill a form field
- These techniques improve the **interaction** and **UX**

CSS Transitions

- CSS supports wide range of element **transitions**

```
button {  
    margin: 5px;  
    padding: 8px 16px;  
    border-radius: 5px;  
    box-shadow: 0 4px 6px  
    rgba(0,0,0,0.3);  
    font-weight: bold;  
    color: white;  
    background: blue;  
    border: 1px solid lightblue;  
    cursor: pointer;  
    transition: all 0.3s ease;  
}
```

```
button:hover {  
    background: darkblue;  
    scale: 1.05;  
}  
  
button:active {  
    scale: 0.95;  
    transform: translateY(-2px);  
}
```

Buttons with transitions:

Yes

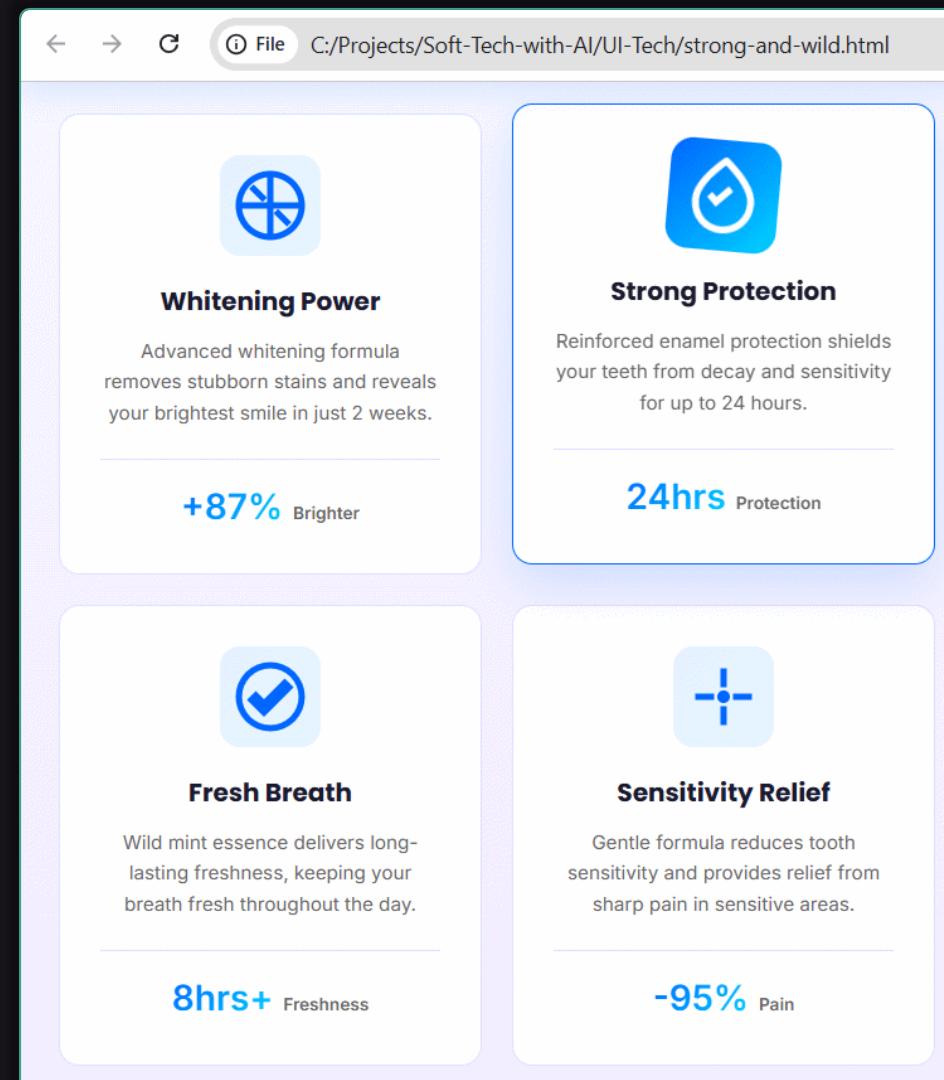
No

```
<p>Buttons with transitions:</p>  
<button>Yes</button>  
<button>No</button>
```

Exercise: Modern Site

Build a **modern Web site** about a **toothpaste** "Strong and Wild"

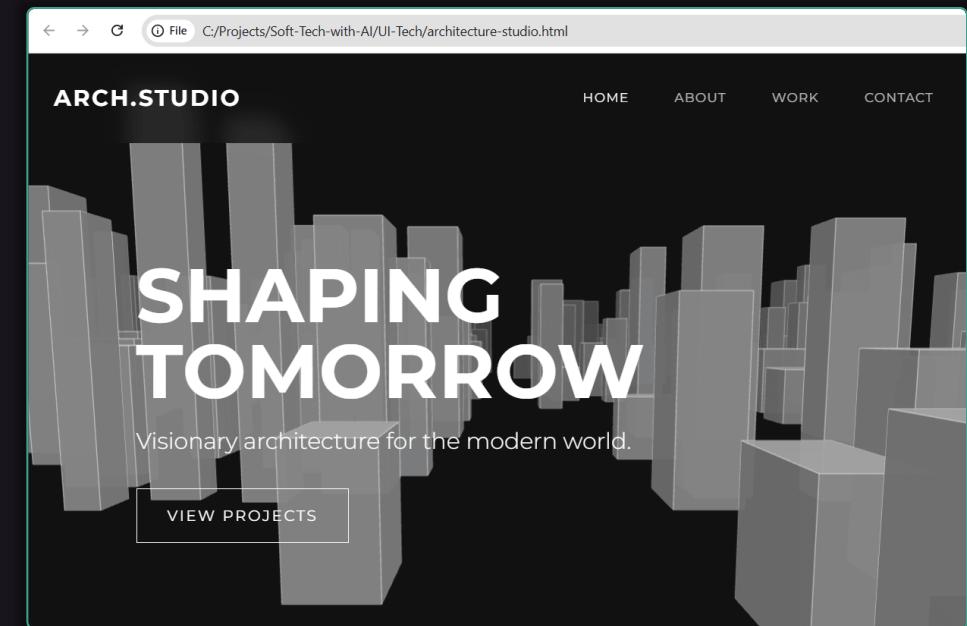
- Use HTML, CSS and **Bootstrap**
- Single page with navigation and section scrolling: **toothpaste.html** + **toothpaste.css**
- Sections: **Product** | **Benefits** | **Order Now**
- Combine text, graphics, icons, lotties for modern visual communication
- Act as professional interaction designer: add **UI transitions, effects** and **microinteractions** to enhance the visual communication



Three.js: 3D Graphics and Animation

- Three.js is in-browser **3D graphics and animation** library
 - Use it to create 3D visualizations, apps and games
- Example: run the **prompt** with a **powerful AI model**

Using Three.js and Bootstrap, create a modern single page site for architectural studio. In the site hero section play a creative 3D animation: camera flying between buildings.



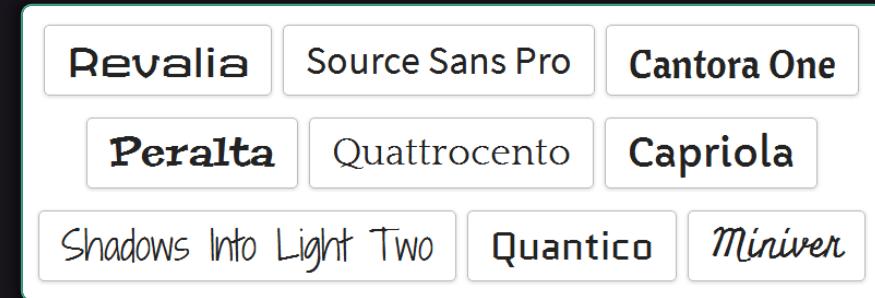
Web Fonts

Google Fonts Library, Using Web Fonts



Web Fonts

- **Web fonts** are fonts hosted online, loaded by the browser at runtime
 - Improve site's look and feel
- **Font libraries** provide free (and paid) fonts for Web apps
- **Google Fonts** – the most widely used online font platform
 - Provides 2000 free fonts, many languages, many styles
 - Download a local copy of the font or use it from a CDN
 - <https://fonts.google.com>



A screenshot of a web browser displaying the Google Fonts website at fonts.google.com/. The page shows a live demo of the text "Hello Google Fonts!" in various font styles. On the left, a sidebar provides options for previewing different categories: Fonts (selected), Noto, Icons, Knowledge, and FAQ. A preview window shows the text "Hello Google Fonts!" in a large, bold font. Below it, a font size slider is set to 42px. The main content area features three examples of the text "Hello Google Fonts!" in different fonts: Ubuntu (8 styles by Dalton Maag), Rubik (Variable (2 axes) by Hubert and Fischer, Meir Sadan), and Roboto Slab (Variable (1 axis) by Christian Robertson). Each example includes a "Filters" button and a "Clear filters" link. The top right corner of the browser window has a pink overlay.

Google Fonts

Live Demo

<https://fonts.google.com>

System Fonts

- Web browsers have **system fonts**, coming from the OS:
 - **Windows**: Segoe UI, Arial, Consolas, Courier New
 - **macOS**: San Francisco, Helvetica Neue, ui-monospace
- Use **system fonts with fallback**:

```
body {  
  font-family: ui-monospace, SFMono-Regular, Menlo, /* macOS */  
  Consolas, "Liberation Mono", "Courier New", /* Windows */  
  monospace; /* Universal fallback */  
}
```

Web Fonts (Load from CSS)

- Importing a Web font from CSS:

```
@import  
url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@  
300;600&display=swap');  
/* display=swap → Swap 'sans-serif' with 'Open Sans' when loaded */  
  
body {  
    font-family: 'Open Sans', sans-serif;  
}
```

```
<html><body>  
    <h1>Open Sans from Google Fonts</h1>  
</body></html>
```



Web Fonts (Load from HTML)

- Importing a Web font from HTML:

```
<html>
<head>
  <link rel="stylesheet" href=
  "https://fonts.googleapis.com/css2?family=Oswald:wght@700&
  family=Inter:wght@400;500&display=swap">
</head>
<body>
  <h1 style="font-family: Oswald">Oswald from Google Fonts</h1>
  <p style="font-family: Inter">Inter from Google Fonts</p>
</body>
</html>
```



Font Weights

- Visual weight hierarchy
 - **Font weights** (300, 400, 700) guide the user's eye
 - **Font-weight: normal** (400) | **bold** (700) | **lighter** | **bolder**

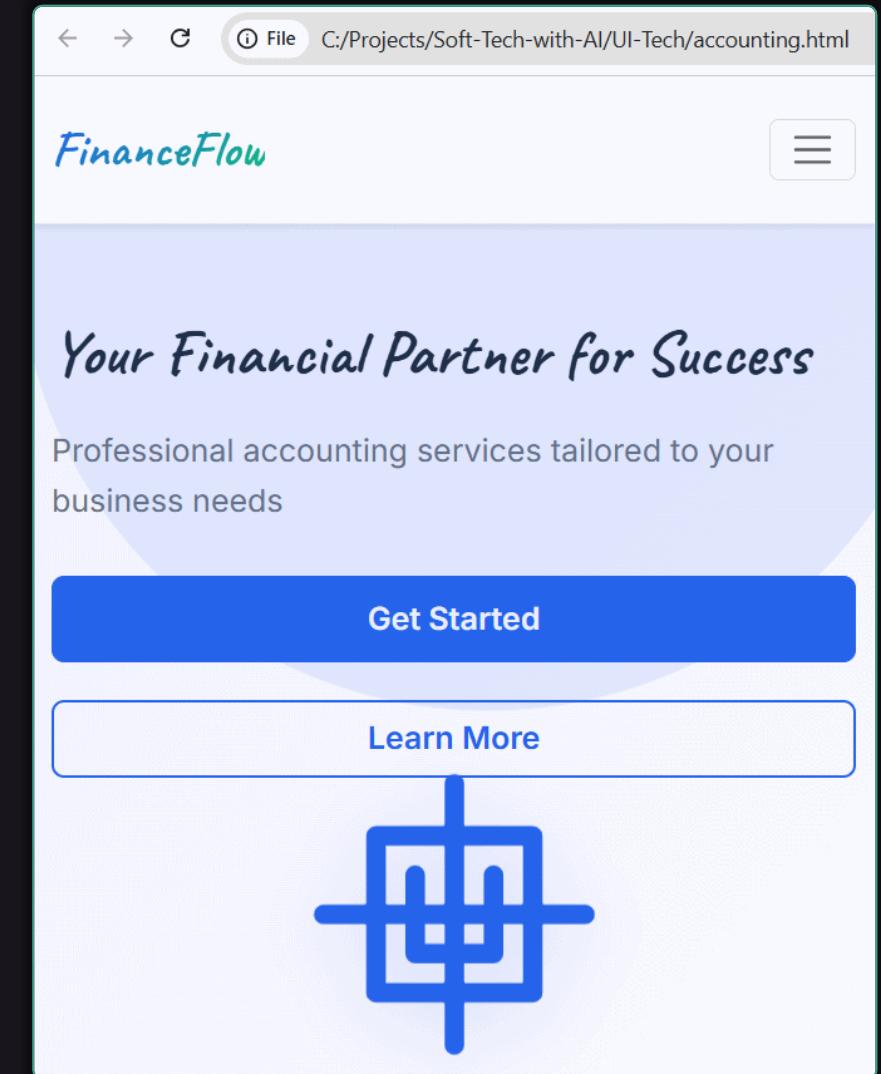
```
body {  
    font-family: Roboto;  
    font-weight: bold;  
}
```

<code>font-weight: 100;</code>	Thin
<code>font-weight: 200;</code>	Extra-Light
<code>font-weight: 300;</code>	Light
<code>font-weight: 400;</code>	Regular
<code>font-weight: 500;</code>	Medium
<code>font-weight: 600;</code>	Semi-Bold
<code>font-weight: 700;</code>	Bold
<code>font-weight: 800;</code>	Extra-Bold
<code>font-weight: 900;</code>	Black

Company Website with Fonts

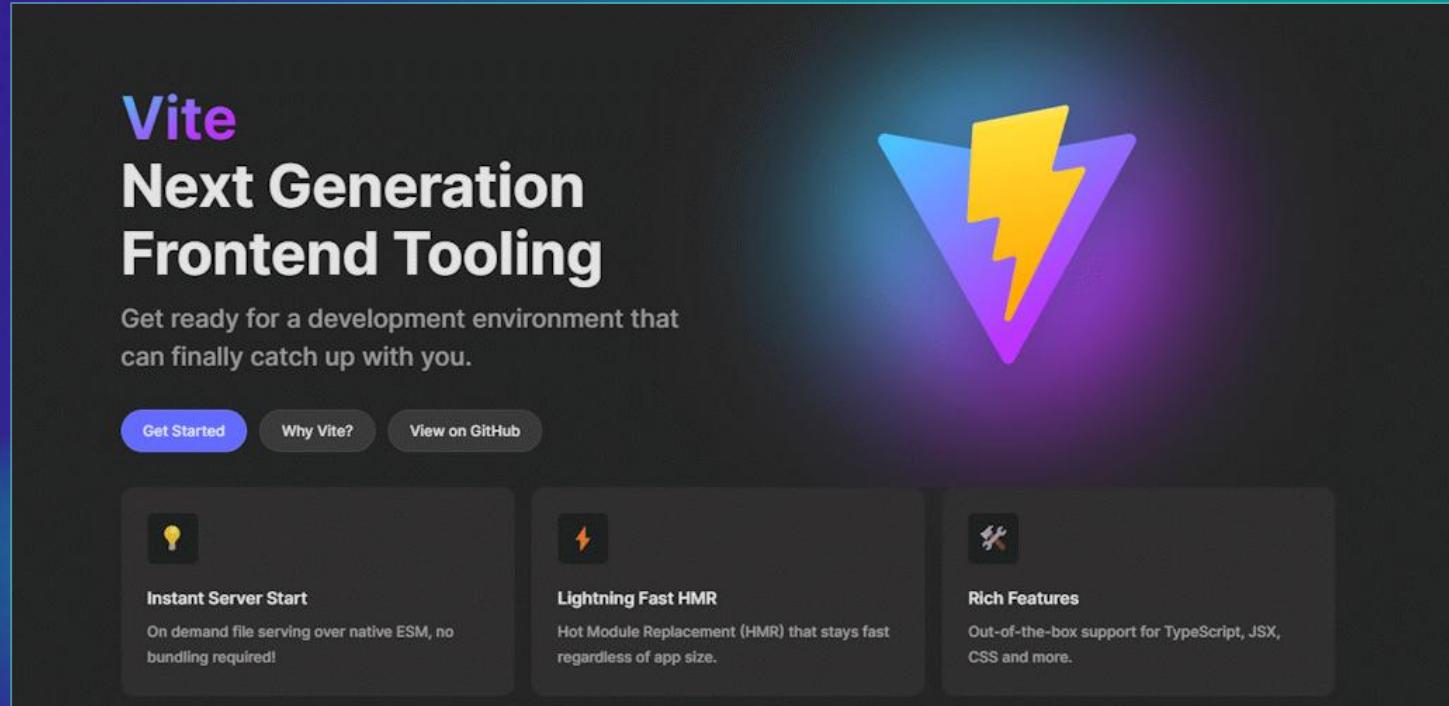
Accounting service company:

- Create a modern **company website** using **Bootstrap + Lucide icons + Google Fonts**
 - **accounting.html, accounting.css**
 - Single page, with sections: **Home, Services, About, Team, Contacts**
 - Use modern combination of **Web fonts**
 - Use a **handwritten font** for titles + site logo
 - Use **icons** to make the content more visual
 - Insert appropriate **favicon**
 - Implement **animations** for better UX



Vite Dev Server

Intro to Vite, Setup Vite Project,
Project Structure, Multi-Page Apps



What is Vite?

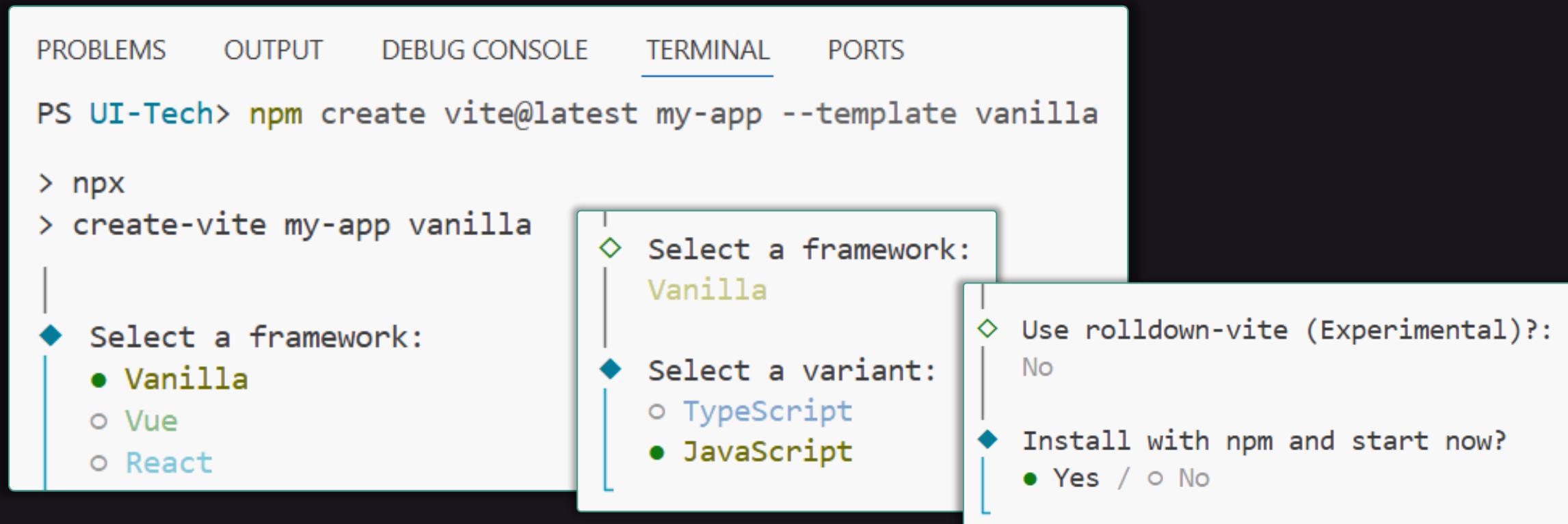
- To use HTML fragments, TypeScript, React, and Tailwind in a Web project, you need a **front-end build tool**
- **Vite** is a **dev server** and **build tool** for Web projects
 - **Serves your code with HTTP** to Web browsers
 - Transforms **TypeScript** into JavaScript on demand
 - Allows **splitting large projects into components** (multiple HTML files, JS files, and CSS files) → **multi-page apps**
 - **Hot reload** on change (without restarting the server)
 - Can **build** static HTML files + scripts for deployment



Creating a Project in Vite

- Creating a Vite project (HTML + JavaScript, no framework):

```
npm create vite@latest my-app --template vanilla
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PS UI-Tech> npm create vite@latest my-app --template vanilla
```

```
> npx
> create-vite my-app vanilla
```

```
|
```

```
◆ Select a framework:
  • Vanilla
  ○ Vue
  ○ React
```

```
|
```

```
◆ Select a variant:
  ○ TypeScript
  • JavaScript
```

```
|
```

```
◆ Use rolldown-vite (Experimental)?:
  No
```

```
|
```

```
◆ Install with npm and start now?
  • Yes / ○ No
```

Running a Vite Project

- Running a Vite project:

```
cd my-app
```

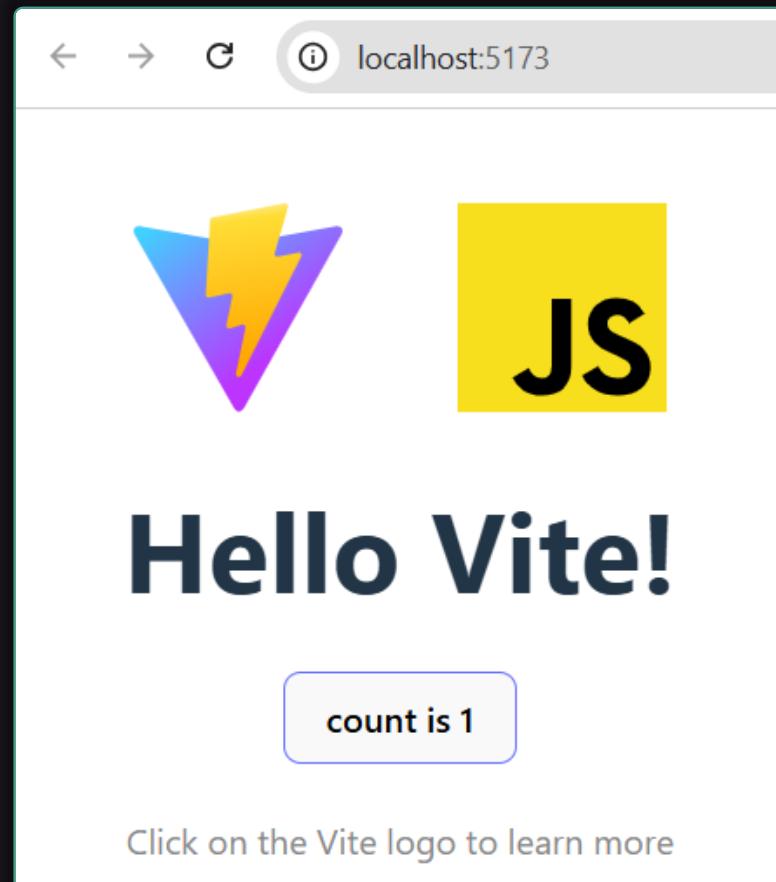
```
npm install
```

```
npm run dev
```

```
VITE v7.3.1 ready in 195 ms
```

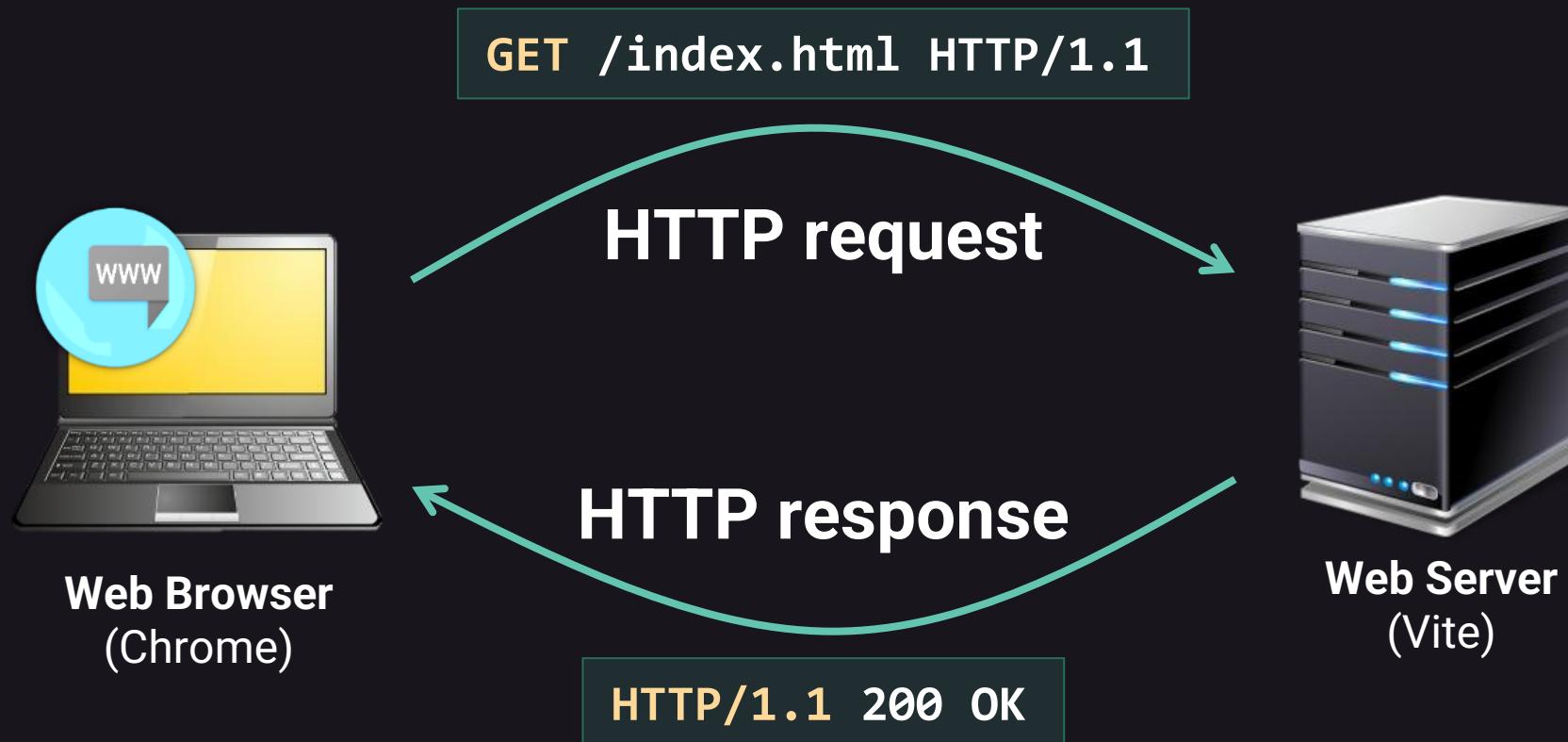
```
→ Local: http://localhost:5173/  
→ Network: use --host to expose  
→ press h + enter to show help
```

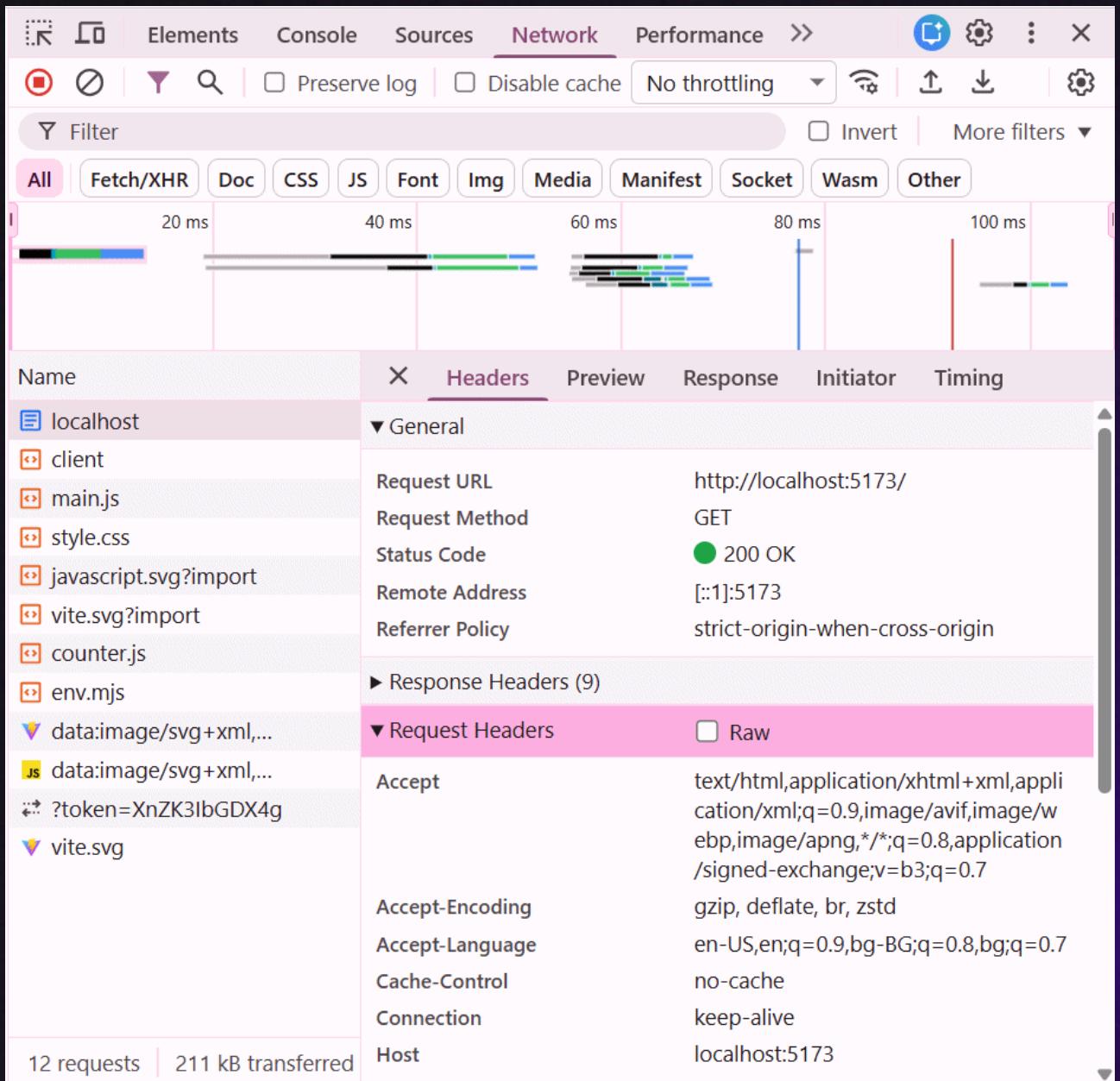
```
my-app
  node_modules
  public
    vite.svg
  src
    counter.js
    javascript.svg
    main.js
    style.css
  .gitignore
  index.html
  package-lock.json
  package.json
```



What is a Web Server?

- **Web servers** serve Web content (HTML, CSS, JS, images) to Web browsers using the **HTTP** protocol



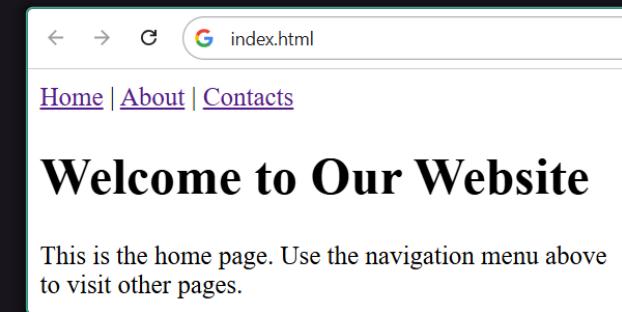
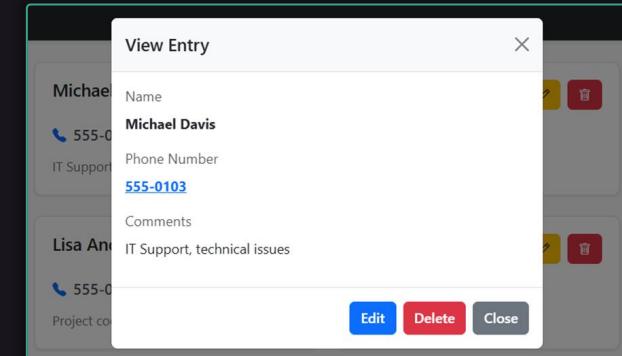


Web Browser, Web Server and HTTP

Live Demo

Multi-Page UI vs. Dialog-Based UI

- **Dialog-based UI** (modal popups)
 - Main page + popup dialogs
 - The entire app code is a **single HTML file** (elements show / hide with user interactions)
- **Multi-page app** (modular design)
 - Split the app into **pages** and **components**:
 - **index.html, about.html, contacts.html, styles.css**
 - Each page opens from its own URL: **/, /about, /contacts**

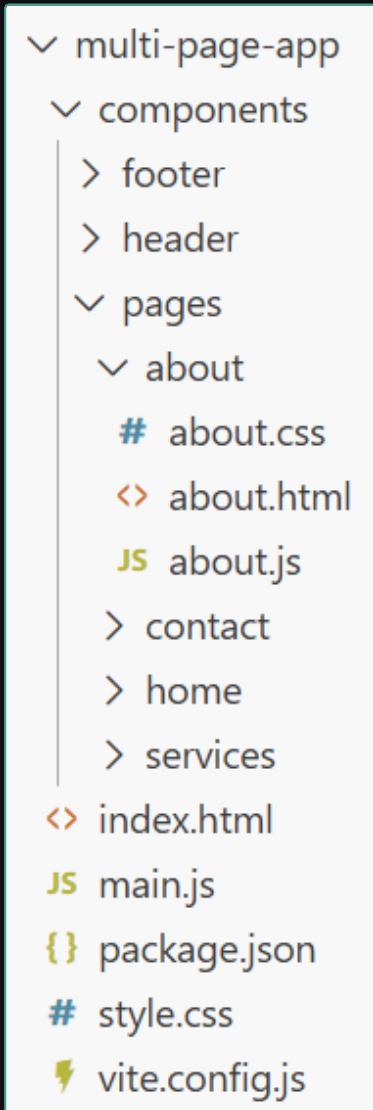


Why Multi-Page Apps?

- **Multi-page UI** vs. **dialog-based UI** (modal popups)
 - **Dialog-based UI** → use for very small apps → too much code in a single file → hard to maintain
 - Why **multi-page**? Experienced developers know → modular structure, easier to maintain, better for bigger apps
- **Modular design** (component-based apps)
 - The app is composed of **modular components**, each with a **well-defined responsibility** and **minimal dependencies**
 - Each component in **separate file** → single responsibility → less code, less complexity → less bugs, easier to maintain

Multi-Page Apps with Vite

- Vite is a **Web server** → can serve app pages from different source code files (HTML, JS, CSS)
- Sample **modular app structure**:
 - **index.html + main.js** → app starting point
 - **Header and footer**:
 - **/components/header/header.html, header.js, header.css**
 - **/components/footer/footer.html, footer.js, footer.css**
 - **Pages**: home, about, contact
 - **/components/pages/home/home.html, home.js, home.css**



Example: Multi-Page App with Vite



Create a **multi-page app** with HTML, Bootstrap and JavaScript

- Setup **Vite** project in VS Code: it will serve the app
 - **Generate all needed Vite files**, don't use a starter kit
- Setup a **multi-page app** with **navigation**
 - **Pages**: Home, Services, About, Contact
 - Implement navigation between pages
 - Change the browser URL when the active page changes
 - Use clean **URLs without hashes**, e.g. <http://localhost:5000/services>
- Structure the app into UI components: **header**, **footer** and **pages**
 - Use HTML fragments for each UI component
 - Use a folder with **separate HTML, CSS and JS file** for each UI component
- Define short **dev agent instructions** (`.github/copilot-instructions.md`)
 - Insert inside appropriate instructions about the app structure

Example: Multi-Page App with Vite

localhost:5173

Multi-Page App

Home Services About Contact

Welcome Home

Your multi-page application built with Vite and Bootstrap

 **Fast Performance**
Lightning-fast development and build times with Vite.

 **Component-Based**
Modular architecture with reusable components.

 **Bootstrap 5.**
Beautiful, responsive design using Bootstrap 5.

Getting Started

This is a multi-page application with client-side routing. Navigate using the menu above to explore different pages. Changes are applied cleanly without page reloads, providing a smooth single-page application experience.

[View Services](#) [Learn More](#) [Contact Us](#)

localhost:5173/services

Multi-Page App

Home Services About Contact

Our Services

Comprehensive solutions for your business needs

 **Web Development**
Custom web applications built with modern technologies and best practices.

- ✓ Responsive Design
- ✓ Progressive Web Apps
- ✓ E-commerce Solutions

 **Mobile Development**
Native and cross-platform mobile applications for iOS and Android.

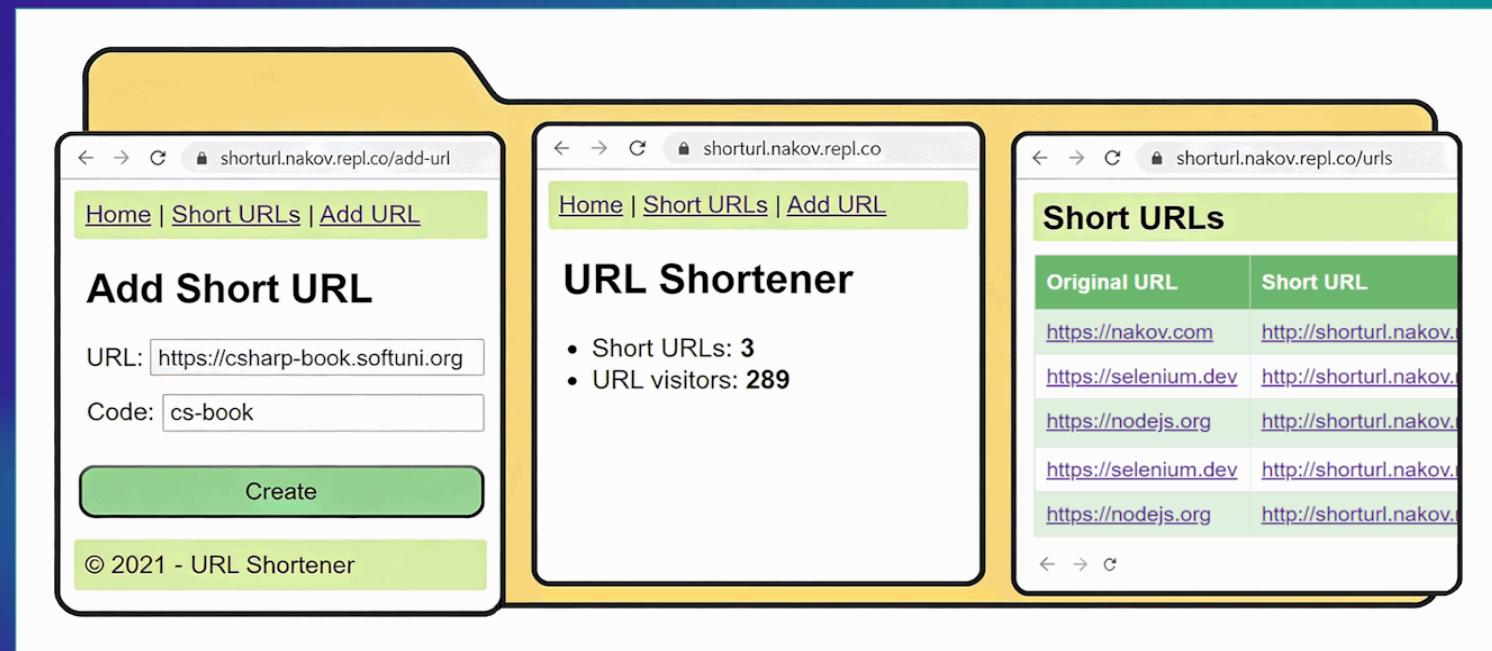
- ✓ iOS & Android Apps
- ✓ Cross-Platform Solutions
- ✓ App Store Optimization

 **UI/UX Design**
Beautiful and intuitive user interfaces that engage your audience.

- ✓ User Research
- ✓ Prototyping
- ✓ Usability Testing

From Image to UI

AI-Generated UI from Images,
Screenshots, Wireframes, Figma Designs



From Image to UI

- AI can generate **code from images**
 - Screenshots, wireframes, Figma designs
- Creating HTML + CSS + JS from **screenshots**
 - Use a simple AI prompt:

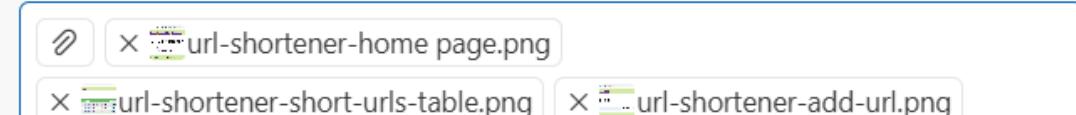
*Create an app that does ...
with a UI similar to the
screenshots*

[Attached screenshots]

Build with Agent

AI responses may be inaccurate.

[Generate Agent Instructions](#) to onboard AI onto your codebase.



Create a "URL Shortener" app in plain HTML + CSS + JS

- Keep the URLs in the browser local storage
- Add a few sample URLs initially
- Implement "Home" page, "Short URLs" page, "Add URL" page
- It should look like at the screenshots (layout, fonts, colors, styles)
- Keep app data in the browser local storage

Agent ▾ Auto ▾  

Exercise: App from Screenshots

Create a "**URL Shortener**" app in plain HTML + CSS + JS (see the attached screenshots)

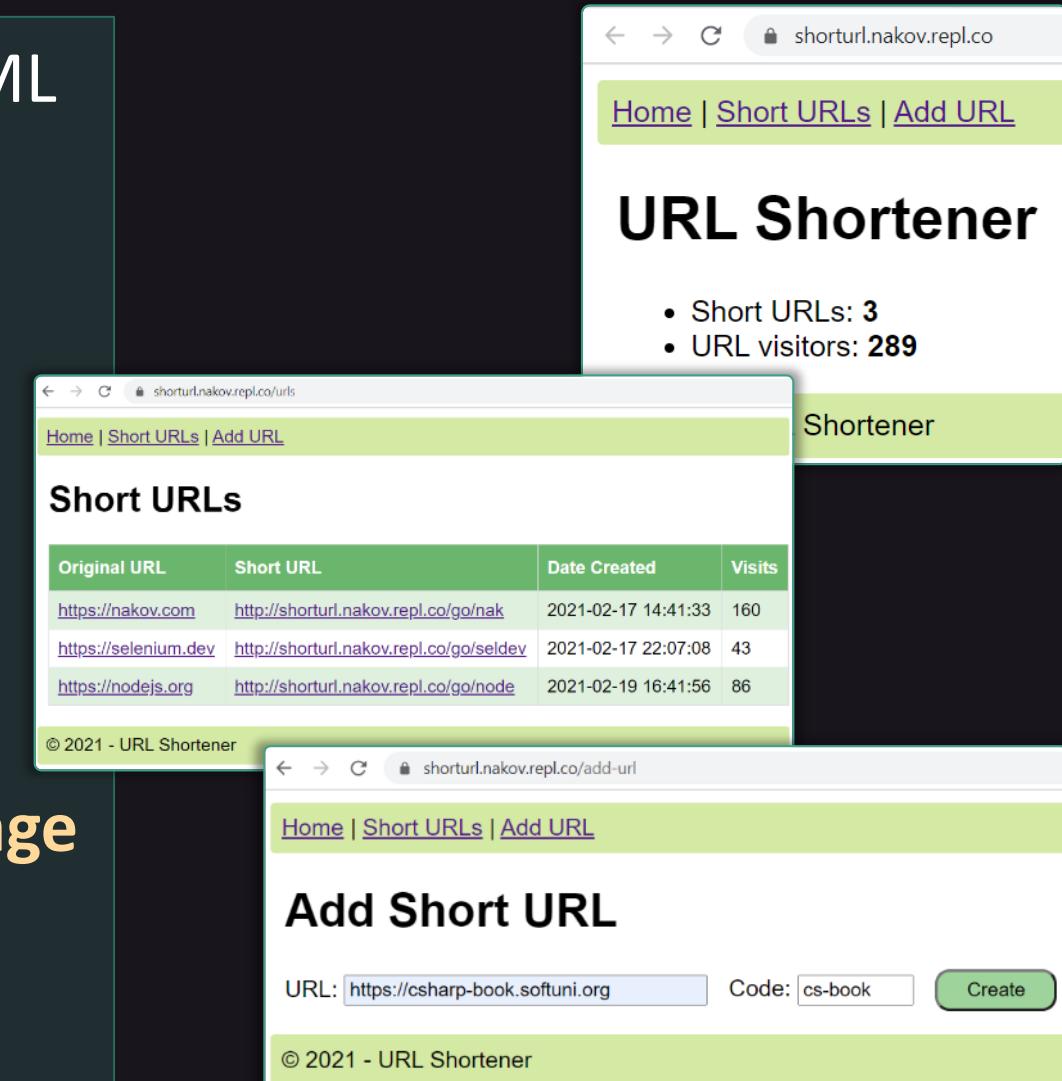
- Implement "**Home**" page, "**Short URLs**" page, "**Add URL**" page

- It should look like at the **screenshots** (layout, fonts, colors, UI styles, URLs)

- The app keeps mapping between **short codes** and **URLs + redirects** code to URL

- Keep app data in the browser **local storage**
- Initially add a few **sample URLs + codes**

[Attached screenshots]



Original URL	Short URL	Date Created	Visits
https://nakov.com	http://shorturl.nakov.repl.co/go/nak	2021-02-17 14:41:33	160
https://selenium.dev	http://shorturl.nakov.repl.co/go/seldev	2021-02-17 22:07:08	43
https://nodejs.org	http://shorturl.nakov.repl.co/go/node	2021-02-19 16:41:56	86

Lesson Summary

- **CSS frameworks** like **Bootstrap**: build responsive UI, using predefined styles and UI components
 - **Bootstrap layout**: containers, rows, columns
 - **UI components**: Navbar, Hero, Card, Accordion, Popup, Drawer, Combobox, Paginator, Toast, Spinner
- Beautiful UI needs: **icons** (Lucide Icons), **Web fonts** (Google Fonts), **lotties**, **SVG** graphics, **animations**
- **Vite** is a development Web server and build tool
 - Can split the app into **modular components** and **pages**

Questions?



Postbank – Exclusive Partner for SoftUni AI



- One of the leading **banking institutions** in Bulgaria
- Member of the Eurobank Group with € 103 billion assets
- Innovative trendsetter with next generation **beyond banking**, transforming today, empowering tomorrow
- Certified **Top Employer 2025** by the international Top Employers Institute
- Proven people care and **wellbeing initiatives**
- Benefits and unlimited access to professional, **personal and leadership trainings and programs**
- www.postbank.bg / careers.postbank.bg



Diamond Partners of Software University



Diamond Partners of SoftUni Digital



**SUPER
HOSTING
.BG**



Diamond Partners of SoftUni Digital



HUMAN

NETPEAK
DIGITAL GROWTH PARTNER



Marmalab | Е-комерс агенция

ETIEN YANEV
Break Your *Limits*. Live Your *Brand*.

1FORFIT



Diamond Partners of SoftUni Creative



Organization Partners of SoftUni Creative



THE
BUCKS
TOWN'S
WORK

