Step-by-Step Guide: Designing & Exporting a Website in Figma

Introduction

This tutorial provides a detailed, step-by-step guide on designing a website in **Figma** and exporting its corresponding **HTML & CSS**. The website will include three key pages:

- **Home Page**
- **About Page**
- **Contact Page**

By following this guide, you will create a structured and professional design that can be easily converted into a functional web layout.

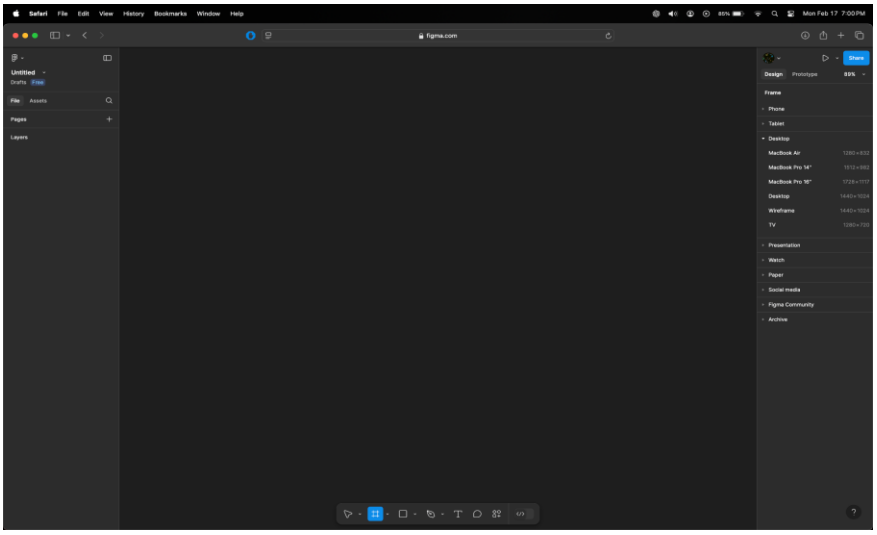
📌 **Figma Template Link:** BUDT748 Figma Example

📌 **Netlify Link:** BUDT748 Deployed Site

---

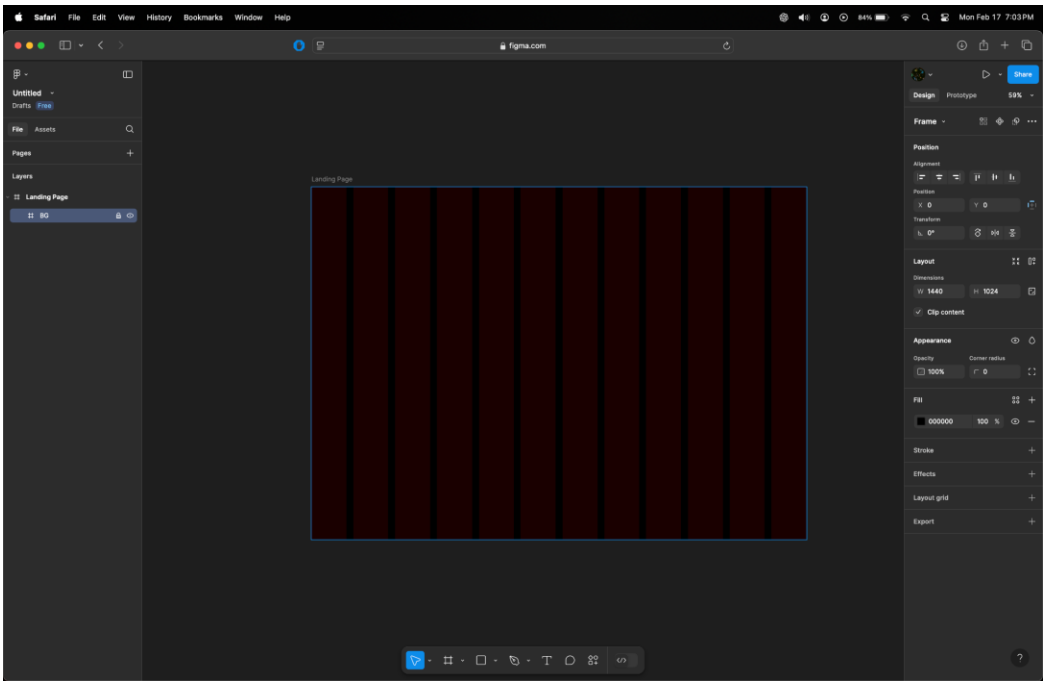
1. Setting Up the Figma Project

1.1 Create a New File & Frame

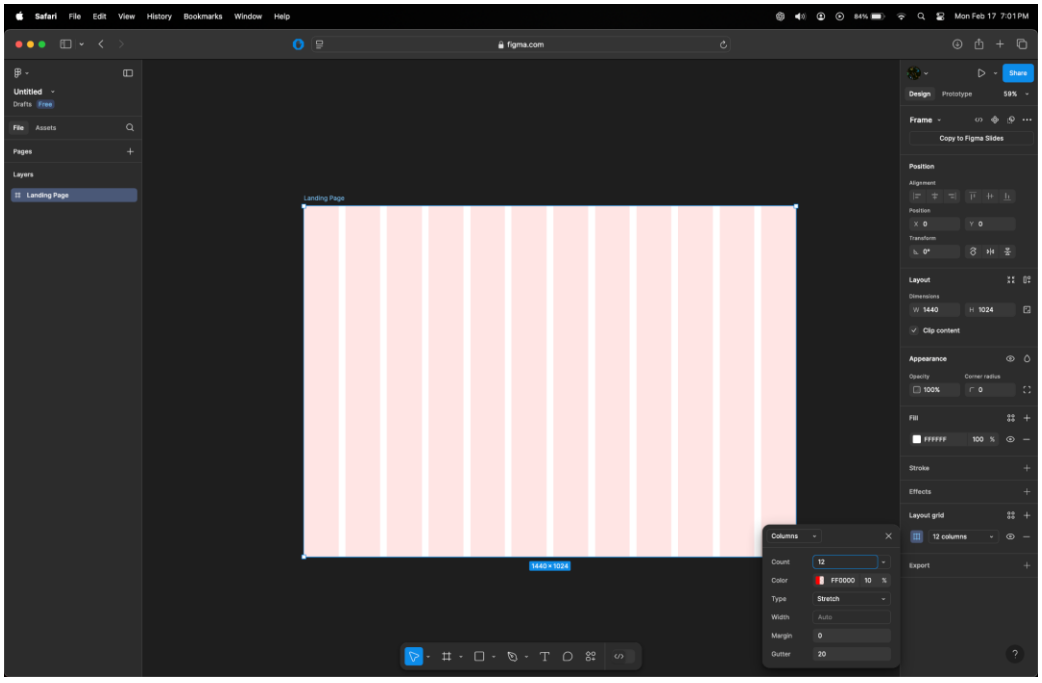


1. Open **Figma** and create a **new design file**.
2. Click on the **Frame Tool (F)** in the toolbar.

3. Select **Desktop (1440px width)** from the right-hand panel.
4. Rename the frame to **Landing Page**.



1.2 Set Up the Grid Layout



1. Select the frame and go to **Layout Grid** in the **Properties Panel**.
2. Change the grid type to **Columns**.
3. Set the following values:
    - **Columns:** 12
    - **Margin:** 80px

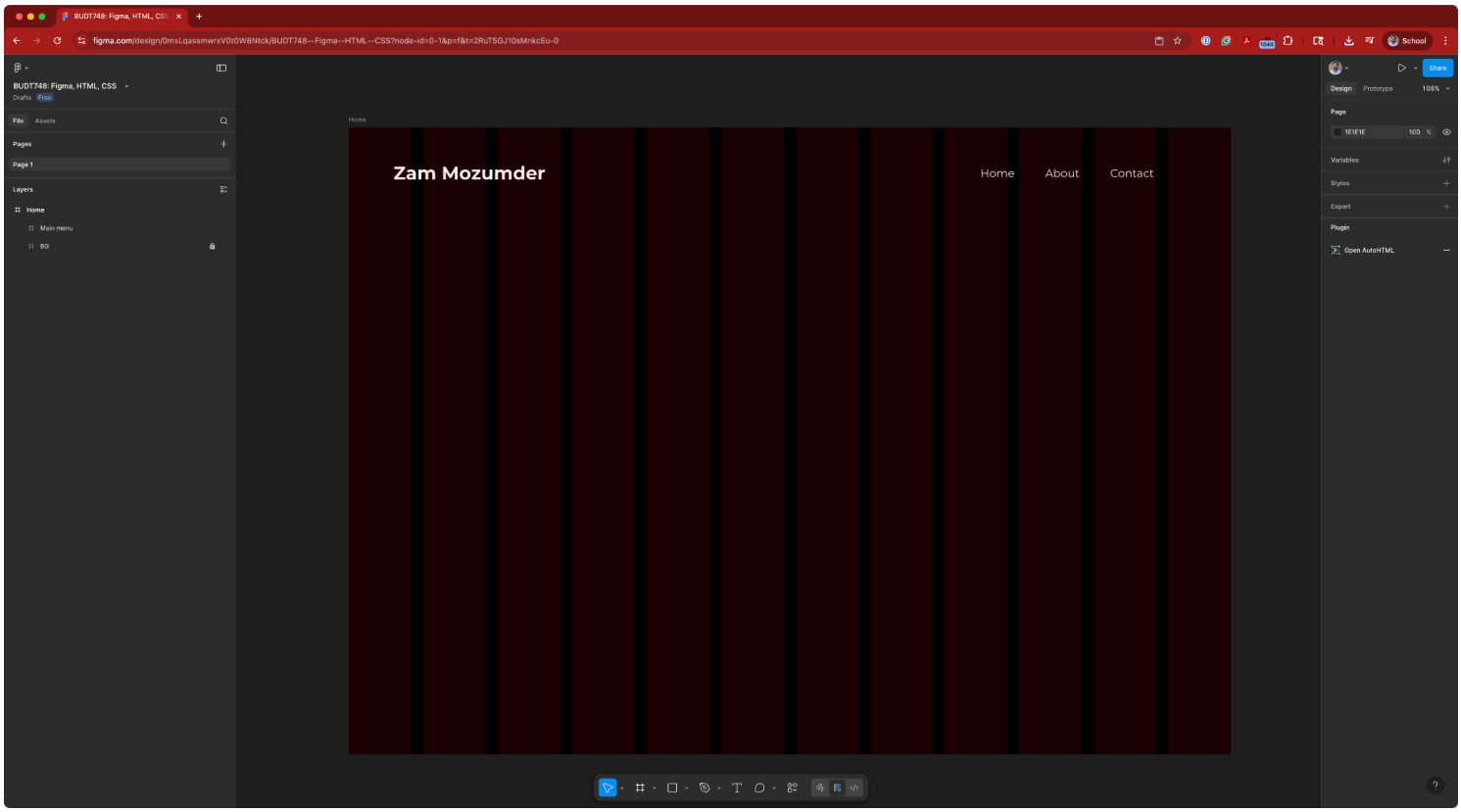
- **Gutter:** 20px

4. This grid will help **align elements properly**.
5. You can toggle the grid visibility using **Ctrl + G (Windows) / Cmd + G (Mac)**.

---

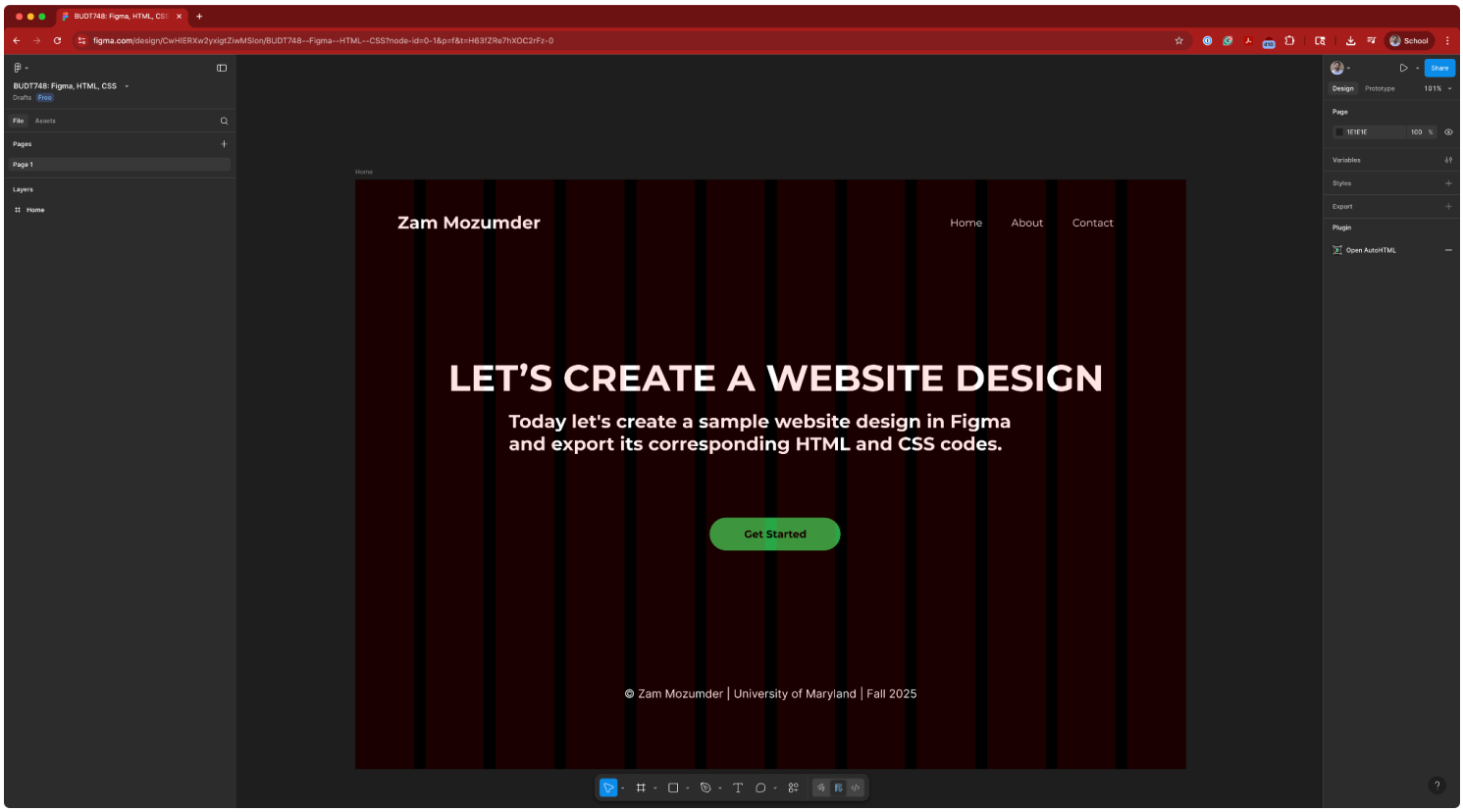
2. Designing the Homepage

2.1 Add the Navigation Bar



1. Use the **Text Tool (T)** to add a **text logo** of your Name.
2. Add **menu items**: Homepage, About, Contact.
3. Apply **Auto Layout (Shift + A)** to space them evenly.
4. Align the navigation to the **top-right corner**.
5. Ensure proper spacing between elements by checking the **grid alignment**.

2.2 Create the Hero Section



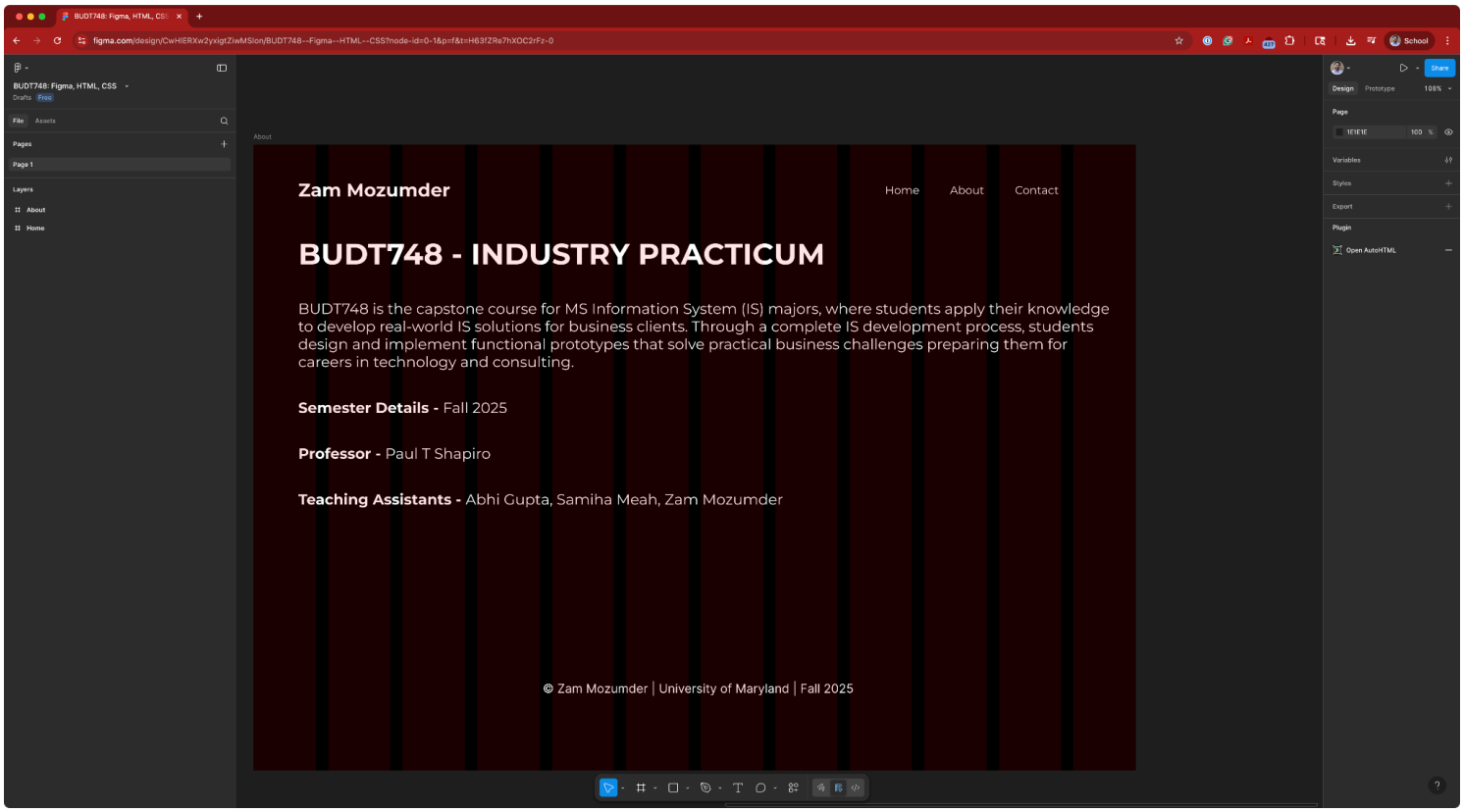
1. Add a large heading using the **Text Tool (T)**:
    - My version has the "LET'S CREATE A WEBSITE DESIGN"
    - Font: **Montserrat Bold, 72px**, White
    - Center-align the text.
2. Add a subtitle below:
    - "Today let's create a sample website design in Figma and export its corresponding HTML and CSS codes."
        i. Replace this with a short blurb about your work experience.
    - Font: **Montserrat Regular, 24px**, White
    - 30px spacing below the heading.
3. Add a **Call-to-Action Button**:
    - Use **Rectangle Tool (R)**, size **180px x 57px**.
    - Fill color: **#A8FF35 (Green)**.
    - Add text: "Get Started".
    - Center-align the button.
    - Apply **drop shadow** for a modern UI effect.
4. Add a Footer using the **Text Tool (T):**
    - My version has: "© Zam Mozumder | University of Maryland | Fall 2025" (replace it with your name)
    - Font: **Montserrat**, **Regular**, **20px**, White
5. Align all elements centrally using **Auto Layout (Shift + A)**.

3. Designing the About Page

3.1 Duplicate the Homepage Frame

1. Select the **Home Frame** and duplicate it (Ctrl + D / Cmd + D).
2. Rename it to **About**

3.2 Modify the Content



1. Change the hero section text:
    - **Title:** "BUDT748 INDUSTRY PRACTICUM"
    - **Subtitle:** Short description about the course.
2. Add **Course Information**:
    - BUDT748 is the capstone course for Information Systems (IS) majors.
    - Some subtext.
3. Add **Professor & TA Details**:
    - **Professor:** Paul T. Shapiro
    - **Teaching Assistants:** Abhi Gupta, Samiha Meah, Zam Mozumder
    - **Semester Details:** Fall 2025
4. Ensure **proper spacing** using **Auto Layout (Shift + A)**.

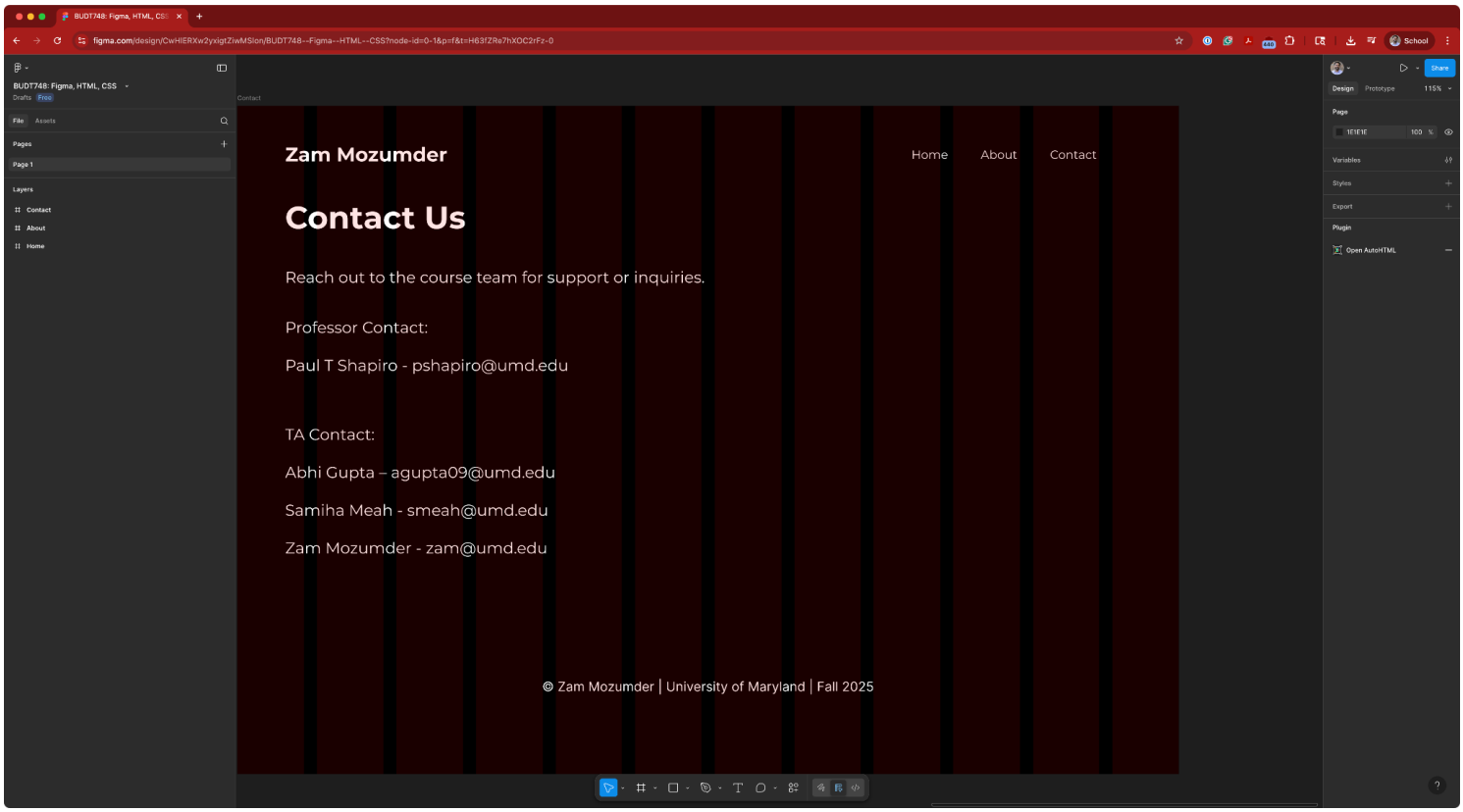
---

4. Designing the Contact Page

4.1 Duplicate the About Frame

1. Select the **About Page Frame** and duplicate it.
2. Rename it **Contact**.

4.2 Modify the Content



1. Change the hero section text:
    - **Title:** "Contact Us"
    - **Subtitle:** "Reach out to the course team for support or inquiries."
2. Add **Professor Contact Details**:
    - **Paul T. Shapiro** – pshapiro@umd.edu
        i. Above Professor Contact - add your name and your contact information.
3. Add **TA Contact Details**:
    - Abhi Gupta – agupta09@umd.edu
    - Samiha Meah - smeah@umd.edu
    - Zam Mozumder - zam@umd.edu
4. Ensure all elements are aligned properly with the **grid system**.

---

5. Exporting & Converting to HTML & CSS

5.1 Exporting the Design

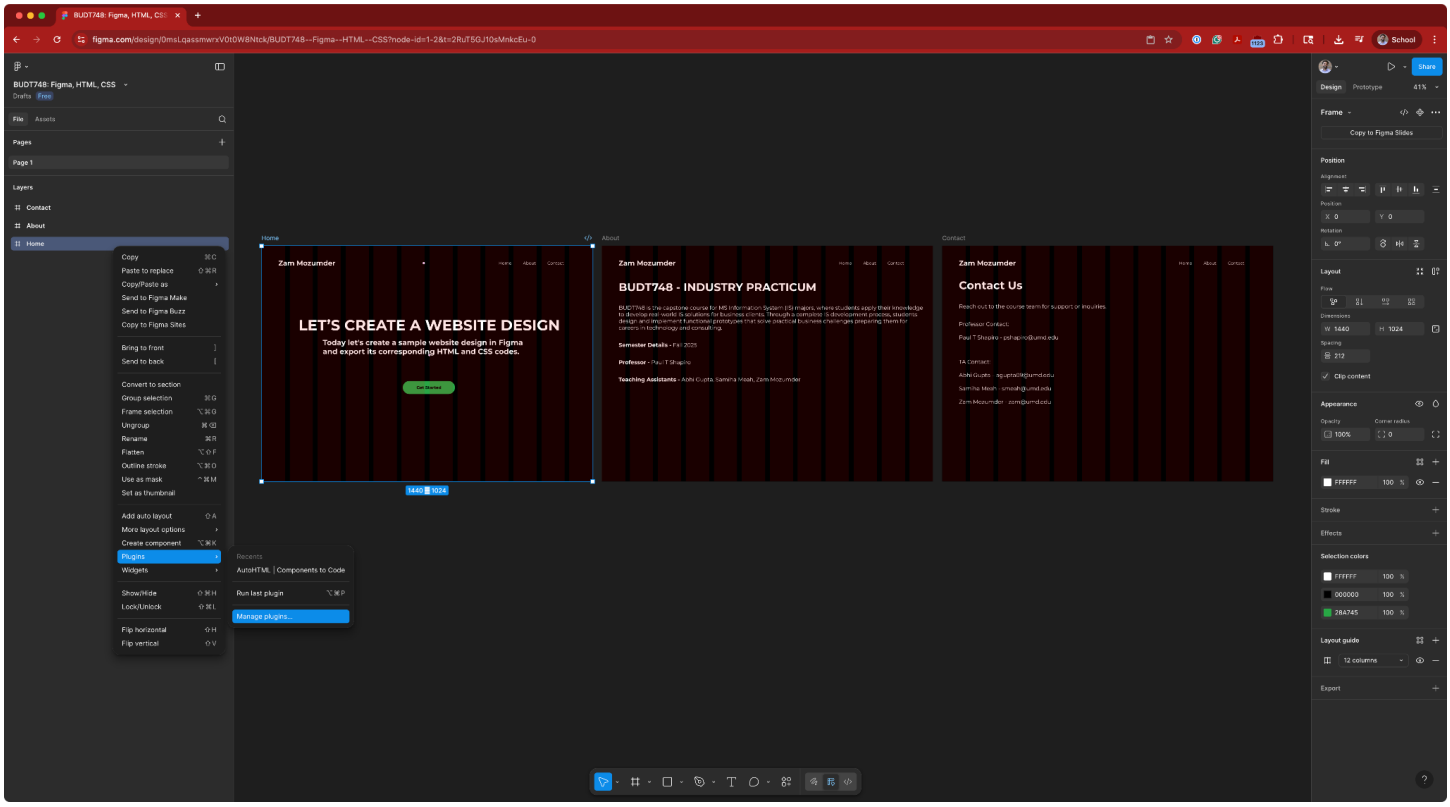
1. Click on any frame.
2. Right-click → **Export**.
3. Choose **PNG, JPG, or SVG** to download assets for use in the website.

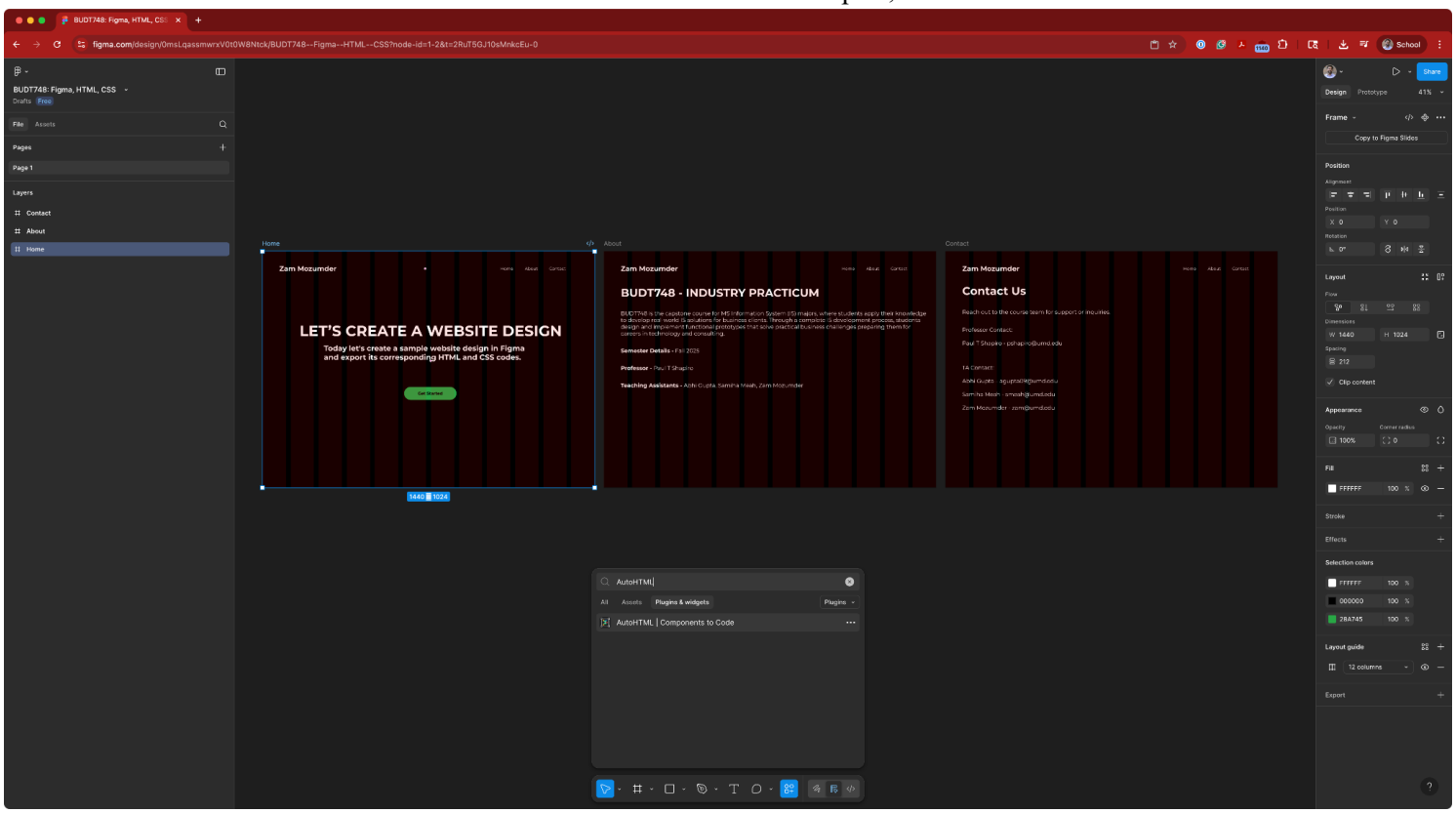
5.2 Converting Design to HTML & CSS

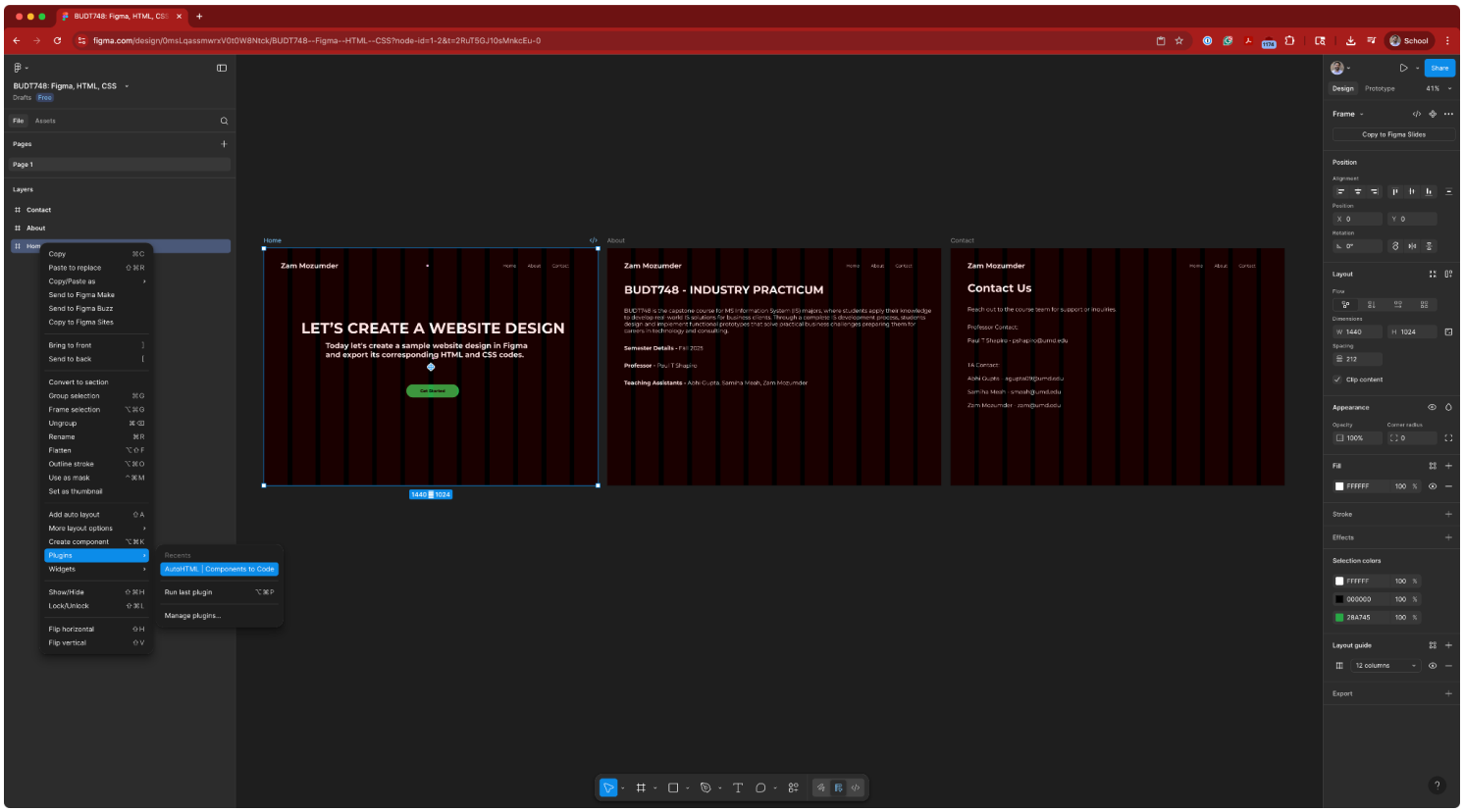
Figma allows you to convert your design into HTML & CSS using free plugins. One such plugin is **AutoHTML | Components to Code**.

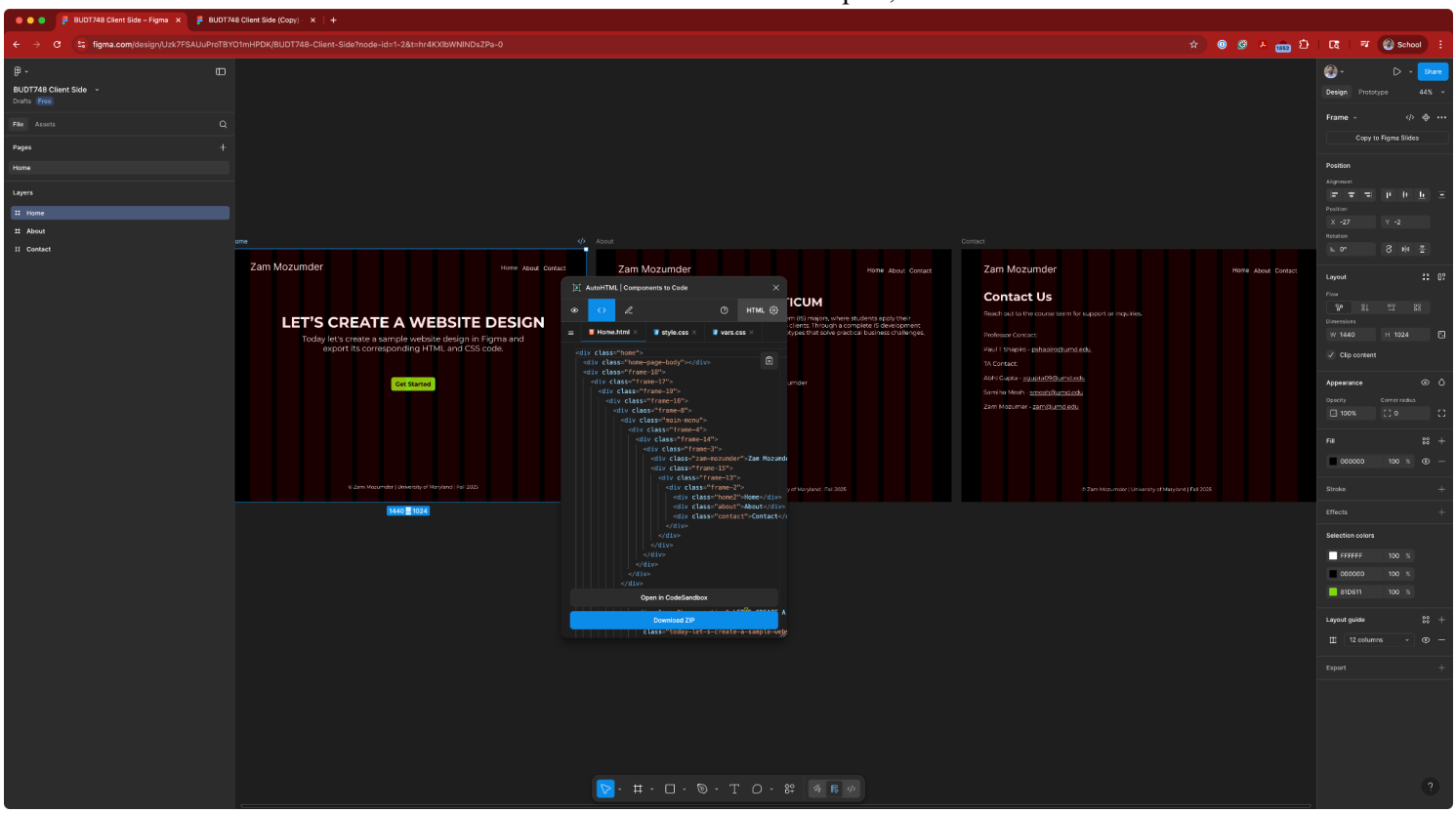
How to Use AutoHTML Plugin:

1. Go to **Figma Plugins**.
2. Search for **AutoHTML | Components to Code** and install it.
3. Select your **components or frame**.
4. Open the **AutoHTML plugin**.
5. Click **Generate Code** to get **HTML & CSS**.
6. You have two options:
    - **Download .zip**: Saves the generated HTML & CSS as a zip file (may not always work properly).
    - **Open in CodeSandbox**: Opens in a new browser tab where you can directly copy the corresponding HTML & CSS code.
7. Use the generated **HTML & CSS files** instead of creating new ones.



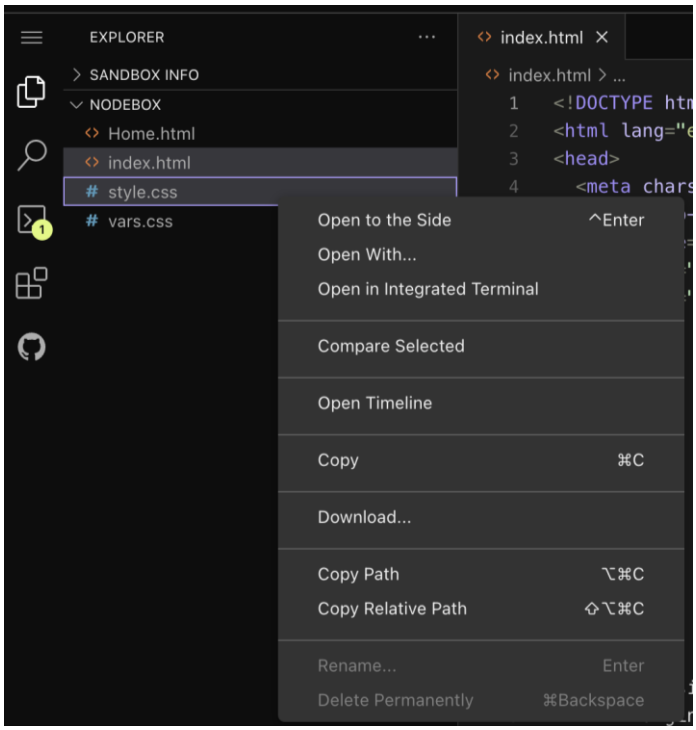

BUDT748: Figma, HTML, CSS
figma.com/design/0msLqassmwrxV0t0W8Ntck/BUDT748--Figma--HTML--CSS?node-id=1-2&t=2RuT5GJ10sMnkcEu-0
School
Share
Design
Prototype
41%
Frame
Copy to Figma Slides
Position
Layout
Appearance
Fill
Stroke
Effects
Selection colors
Layout guide
Export
Pages
Page 1
Layers
Contact
About
Home
Home
About
Contact
Zam Mozumder
LET'S CREATE A WEBSITE DESIGN
Today let's create a sample website design in Figma and export its corresponding HTML and CSS codes.
Get Started
BUDT748 - INDUSTRY PRACTICUM
Contact Us
1440 1024
AutoHTML
All
Assets
Plugins & widgets
Plugins
AutoHTML | Components to Code


BUDT748: Figma, HTML, CSS
figma.com/design/0msLqassmwrxV0t0W8Ntck/BUDT748--Figma--HTML--CSS?node-id=1-2&t=2RuT5GJ10sMnkcEu-0
School
Share
Design
Prototype
41%
Frame
Copy to Figma Slides
Position
Layout
Appearance
Fill
Stroke
Effects
Selection colors
Layout guide
Export
Pages
Page 1
Layers
Contact
About
Home
Copy
Paste to replace
Copy/Paste as
Send to Figma Make
Send to Figma Buzz
Copy to Figma Sites
Bring to front
Send to back
Convert to section
Group selection
Frame selection
Ungroup
Rename
Flatten
Outline stroke
Use as mask
Set as thumbnail
Add auto layout
More layout options
Create component
Plugins
Widgets
Show/Hide
Lock/Unlock
Flip horizontal
Flip vertical
Recents
AutoHTML | Components to Code
Run last plugin
Manage plugins...
Home
About
Contact
Zam Mozumder
LET'S CREATE A WEBSITE DESIGN
Today let's create a sample website design in Figma and export its corresponding HTML and CSS codes.
Get Started
BUDT748 - INDUSTRY PRACTICUM
Contact Us
1440 1024



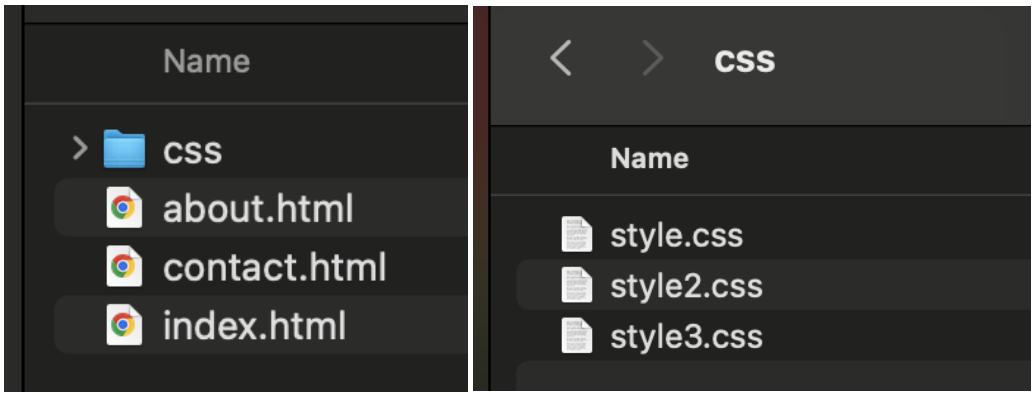
Recommended: Open each page in CodeSandBox one by one and download it independently (make sure you rename the files before you download the next page).

For Home, About, and Contact Pages, download the index.html and style.css files for each, then rename them based on the project structure below:



Project Structure After Exporting:

|— index.html (Homepage (update in VSCode)→ links to css/styles.css)
|— about.html (About Page (Update in VSCode) → must link to css/styles2.css)
|— contact.html (Contact Page (Update in VSCode) → must link to css/styles3.css)
|— css/
| ├— styles.css (Homepage styles)
| ├— styles2.css (About page styles)
| └— styles3.css (Contact page styles)



Understanding HTML & CSS Structure

HTML (HyperText Markup Language) – https://www.w3schools.com/html/

HTML is the standard language for creating webpages. It defines the structure of your page using elements such as headings, paragraphs, images, and links.

CSS (Cascading Style Sheets) – https://www.w3schools.com/css/default.asp

CSS is used to style HTML elements by changing colors, fonts, layouts, and more.

Understanding How CSS Affects HTML

Styling Elements

Example: Changing the background color, text color, and padding of a `<nav>` element.

```
nav {
  background-color: #333;
  color: white;
  padding: 15px;
}
```

This makes the navigation bar **dark gray**, changes the text color to **white**, and adds **15px of padding**.

Layout Control

Example: Adjusting margin and padding for spacing.

```
.container {
  margin: 20px auto;
  padding: 10px;
  width: 80%;
}
```

This centers the container, **adds spacing**, and sets a **fixed width**.

Responsive Design

Example: Using media queries to adjust the layout for mobile screens.

```
@media (max-width: 768px) {
  .container {
    width: 100%;
    padding: 5px;
  }
}
```

When the screen width is below **768px**, the container takes up the **full width** and reduces padding.

Hover & Interactive Effects

Example: Changing button background color when hovered.

```
button:hover {
  background-color: darkblue;
}
```

This changes the button's background color when hovered, improving **user experience**.

---

6. Setting Up Your Development Environment

Step 1: Install VS Code

What is VS Code? Visual Studio Code (VS Code) is a free, lightweight, and powerful code editor from Microsoft. It supports multiple programming languages and comes with extensions that make development faster and easier.

How to Install:

1. Download **VS Code** from here.
2. Follow the installation process for your OS.
3. Open **VS Code** and set it up for **HTML & CSS** development.

Step 2: Install Extensions for Efficiency

Extensions enhance **VS Code's** functionality. Here are some useful extensions:

- **Live Server** – Automatically refreshes the page when you make changes.
- **Prettier** – Helps format your code to keep it neat and readable.

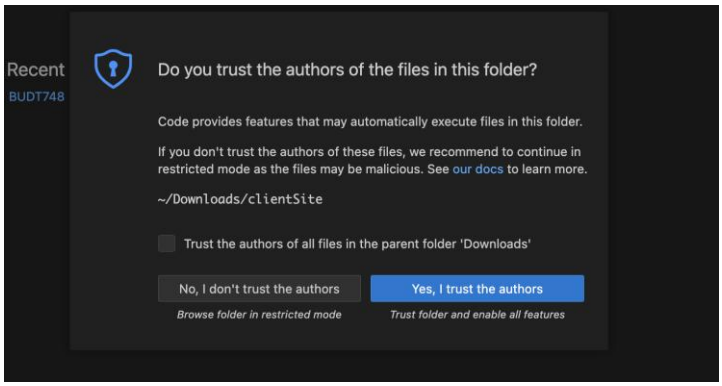
Installation Steps:

- Open **VS Code**.
- Go to **Extensions Marketplace** (Ctrl + Shift + X / Cmd + Shift + X on Mac).
- Search for **"Live Server"** and **"Prettier"**.
- Click **Install**.

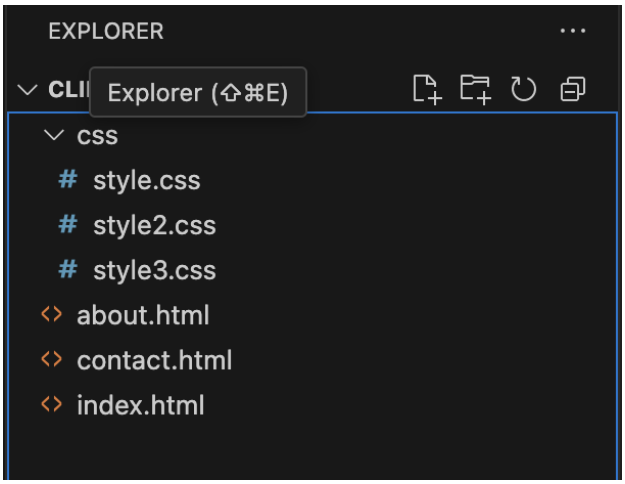
6.1. Editing in VS Code

Installation Steps:

1. Place all the files you download from inside a folder named clientSite.
2. Rename the Stylesheets
3. Open the folder in VS Code (File > Open Folder), Select **Yes, I trust the authors**



4. Your folder structure should be like this:



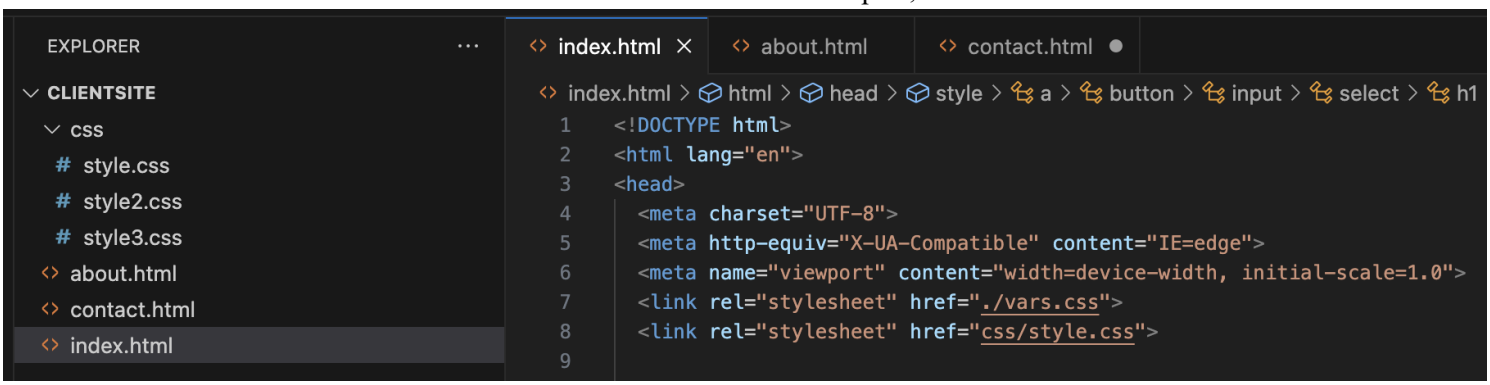
Next, we'll update the HTML and CSS files to ensure each page displays consistently.

1. Each HTML page needs to connect to its matching CSS file. You do this inside the <head> section of your HTML.

- Open index.html, about.html, contact.html
- In the head add this update the style.css to their respective one:
  - index.html, change it to *css/style.css*
  - about.html change it to *css/style2.css*
  - contact.html change it to *css/style3.css*
- Should be on line 8

<link rel="stylesheet" href="css/style.css" />

3. Do this for all 3 pages – you will need to link the style.css, style2.css, style3.css to their respective .html file. It should look like the image below.



---

7. Adding Bootstrap to Your Project

Step 1: What is Bootstrap?

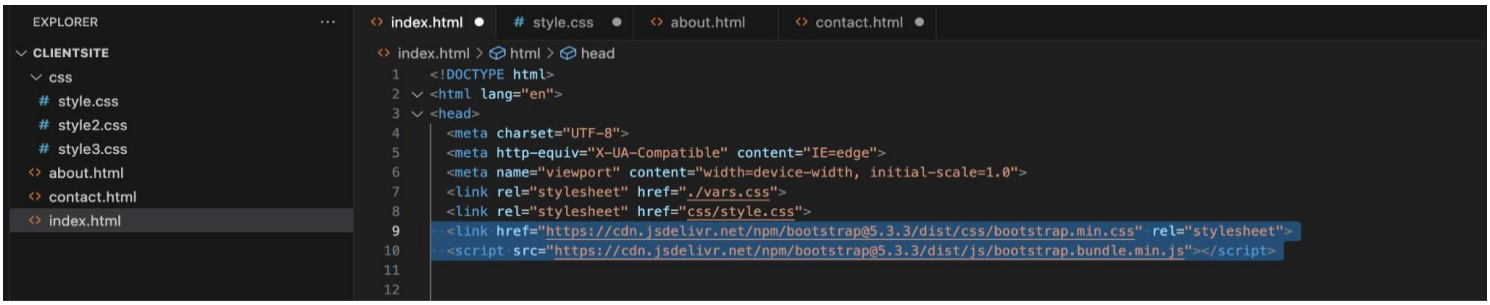
Bootstrap is a popular **CSS framework** that provides **pre-designed styles, responsive layouts, and UI components**, such as buttons, navigation bars, and forms. It makes web development faster and ensures the website looks good on all screen sizes.

Step 2: How to Add Bootstrap

The easiest way to use Bootstrap is to add the following **CDN link** to the `<head>` section of your **HTML file. For every page you use Bootstrap, you will need to add the links (highlighted below in lines 9 and 10)**:

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

This allows you to use Bootstrap classes **without downloading anything manually**.



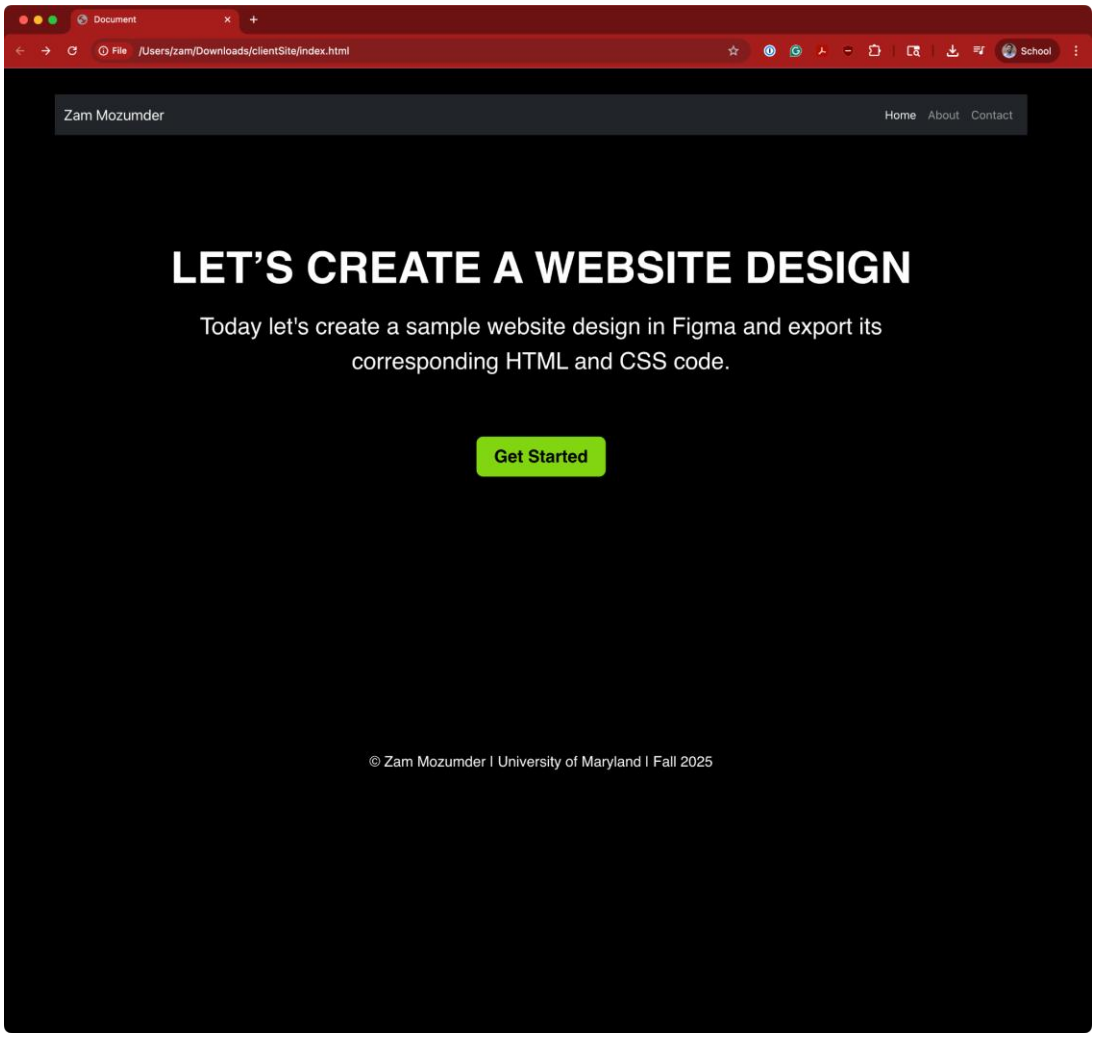
Step 3: Using Bootstrap Components

1. Bootstrap Navigation Bar (Navbar) - Replace the NavBar

```
<div class="main-menu">
```

```
 <nav class="navbar navbar-expand-lg navbar-dark bg-dark w-100">
  <div class="container-fluid">
   <a class="navbar-brand" href="#">Zam Mozumder</a>
   <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
    aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
   </button>
   <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav ms-auto">
     <li class="nav-item"><a class="nav-link active" aria-current="page" href="index.html">Home</a></li>
     <li class="nav-item"><a class="nav-link" href="about.html">About</a></li>
     <li class="nav-item"><a class="nav-link" href="contact.html">Contact</a></li>
    </ul>
   </div>
  </div>
 </nav>
</div>
```

The website should have an updated banner like this:



You will have to remove your existing div class for the Header - these are the following lines I replaced with the navigation bar above. Do it for each of the pages, about, HTML, replace my name with yours, which is within the div class of 'navbar-brand'.

```
      list-style-type: none;
      margin: 0;
      padding: 0;
    }
    </style>
  <title>Document</title>
</head>
<body>
  <div class="home">
    <div class="bg"></div>
    <div class="main-menu">
      <div class="frame-1">
        <div class="home2">Home</div>
        <div class="about">About</div>
        <div class="contact">Contact</div>
      </div>
      <div class="zam-mozumder">Zam Mozumder</div>
    </div>
    <div class="hero-section">
      <div class="let-s-create-a-website-design">
        Let's create a website design
      </div>
      <div
        class="today-let-s-create-a-sample-website-design-in-figma-and-export-its-corresponding-html-and-css-codes"
      >
        Today let&amp;#039;s create a sample website design in Figma
        <br />
        and export its corresponding HTML and CSS codes.
      </div>
      <div class="frame-2">
        <div class="get-started">Get Started</div>
      </div>
    </div>
    <div class="footer">
      <div class="zam-mozumder-university-of-maryland-fall-2025">
        © Zam Mozumder | University of Maryland | Fall 2025
      </div>
    </div>
  </div>

</body>
</html>
```

Feel free to add to the rest of the pages – depending on frames and edits on Figma, it should all be the same lines.

2. Bootstrap Grid System (Responsive Layouts)

Bootstrap's grid system is based on **12 columns** and allows easy layout creation.

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Column 1</div>
    <div class="col-md-6">Column 2</div>
  </div>
</div>
```

- The `col-md-6` class means **two equal columns** on medium and larger screens.
- On smaller screens, Bootstrap **automatically stacks** them.

3. Bootstrap Buttons

<button class="btn btn-primary">Click Me</button>

Bootstrap provides pre-styled buttons like:

- `.btn-primary` (blue)
- `.btn-danger` (red)
- `.btn-success` (green)

4. Bootstrap Forms

```
<form>
  <div class="mb-3">
    <label for="email" class="form-label">Email</label>
    <input type="email" class="form-control" id="email" placeholder="Enter your email">
  </div>
  <button type="submit" class="btn btn-success">Submit</button>
</form>
```

The `form-control` class makes the input fields **responsive and styled**.

---

8. Deploying the Website using GitHub

Step 1: Push Project to GitHub

```
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/your-username/website.git
git push -u origin main
```

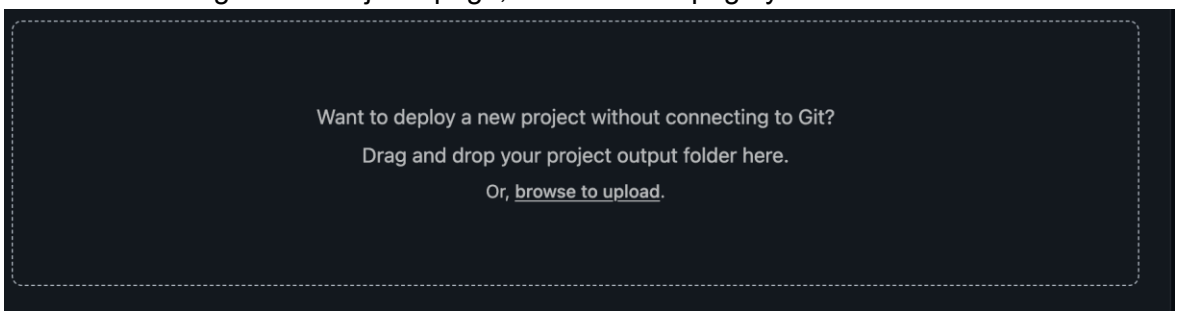
Step 2: Deploy to GitHub Pages

1. Go to **Settings** → **Pages**.
2. Select the **main** branch.
3. Click **Save**.
4. Your site will be live at:
   https://your-username.github.io/website/

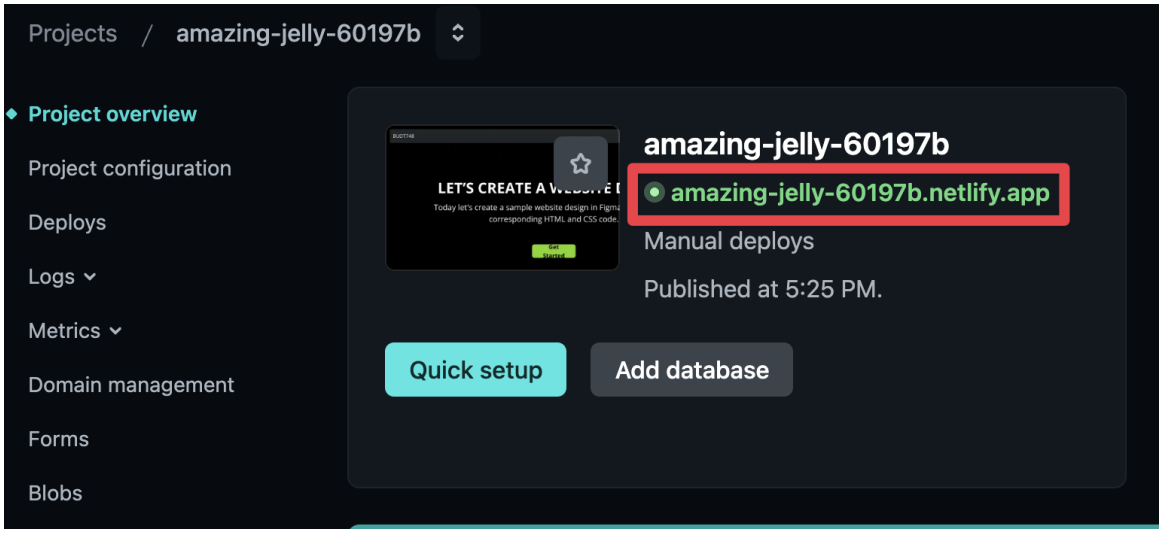
---

9. Alternative Deployment

Step 1: Netlify

1. Visit https://app.netlify.com/login
2. Log in with GitHub or Create an account
3. You will be brought to a Projects page, middle of the page you will see this:
4. 

5. Click Browse to upload and upload the folder where you have your files.
6. Once it's uploaded, it will provide a lik to your site, please submit this link in your assignment.
    a. Links will be verified and source code will be inspected.



Conclusion

You have now built a **website** using **HTML, CSS, Bootstrap, and Figma**.

Next Steps:

- Experiment with **more Bootstrap components** to enhance the website.
- Add more **styling and animations** to improve user experience.
- Learn **JavaScript for dynamic interactions**, such as adding form validation and animations.