



**BICOL UNIVERSITY**

**COLLEGE OF SCIENCE**

Legaspi City



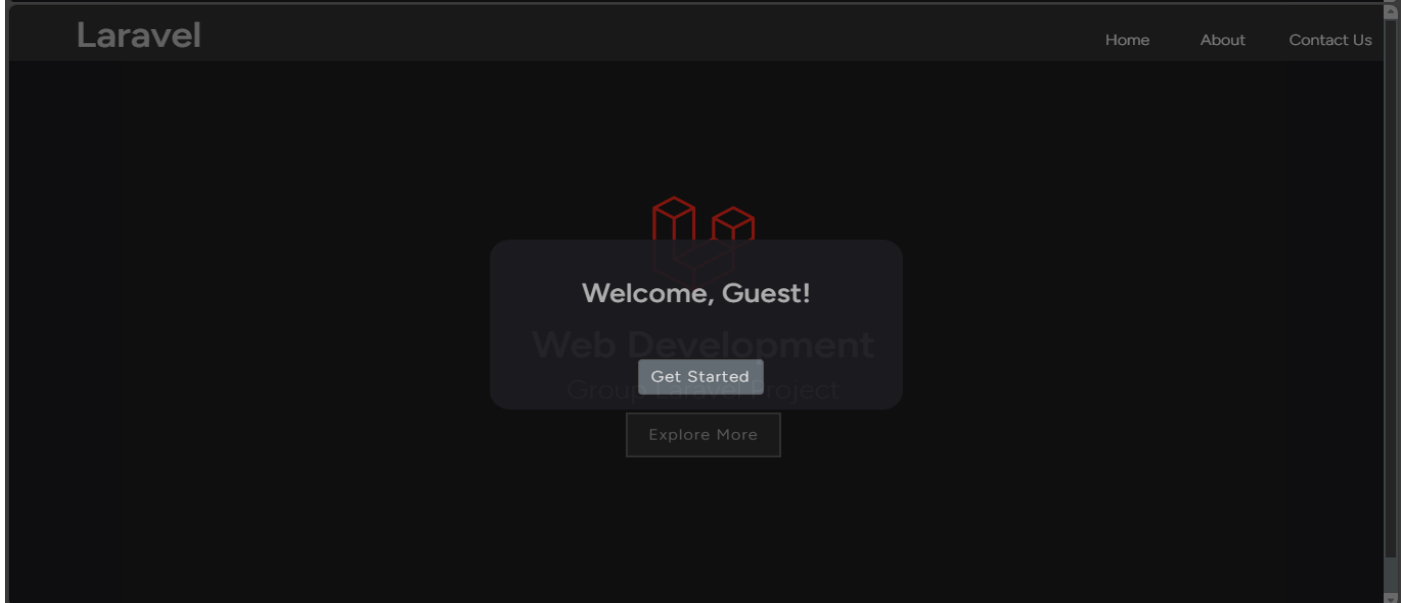
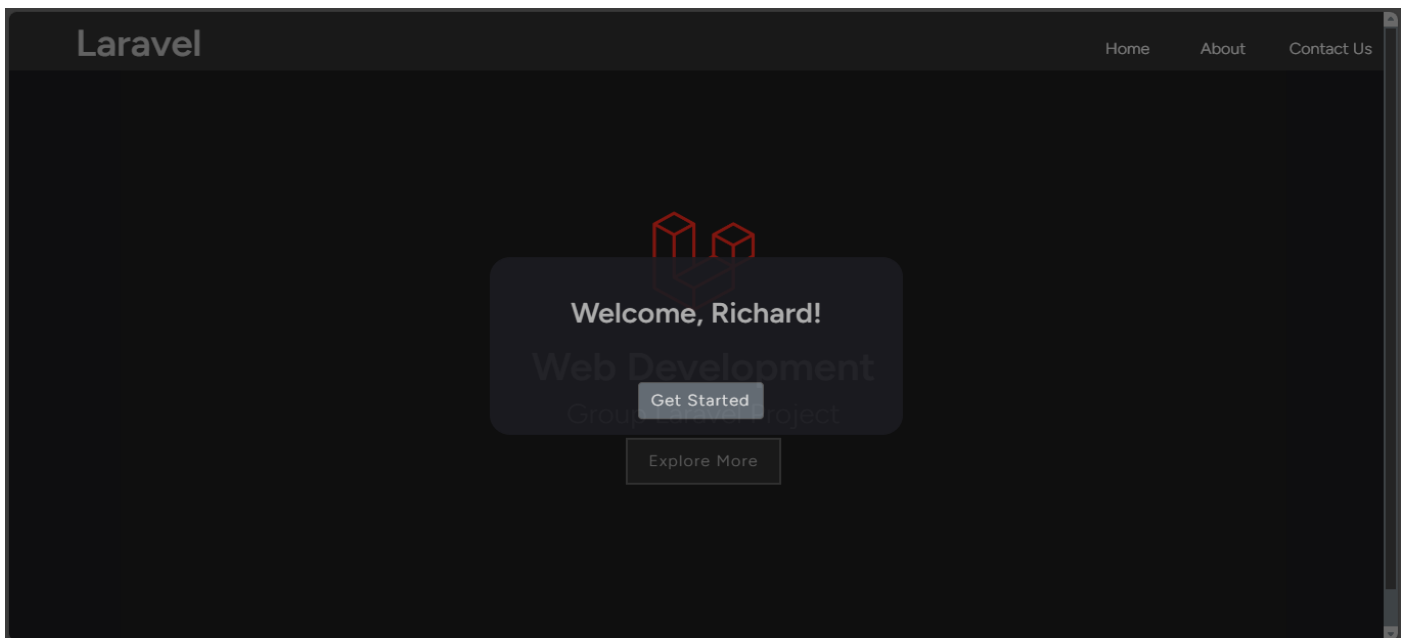
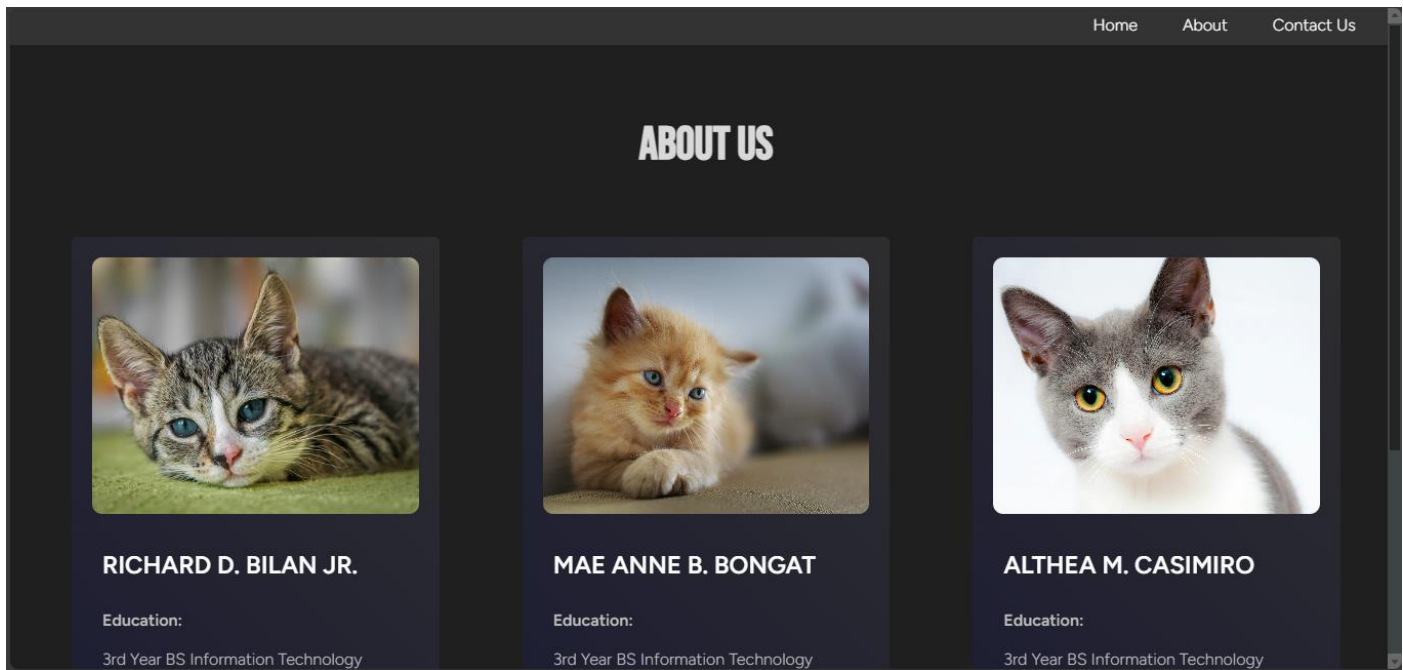
---

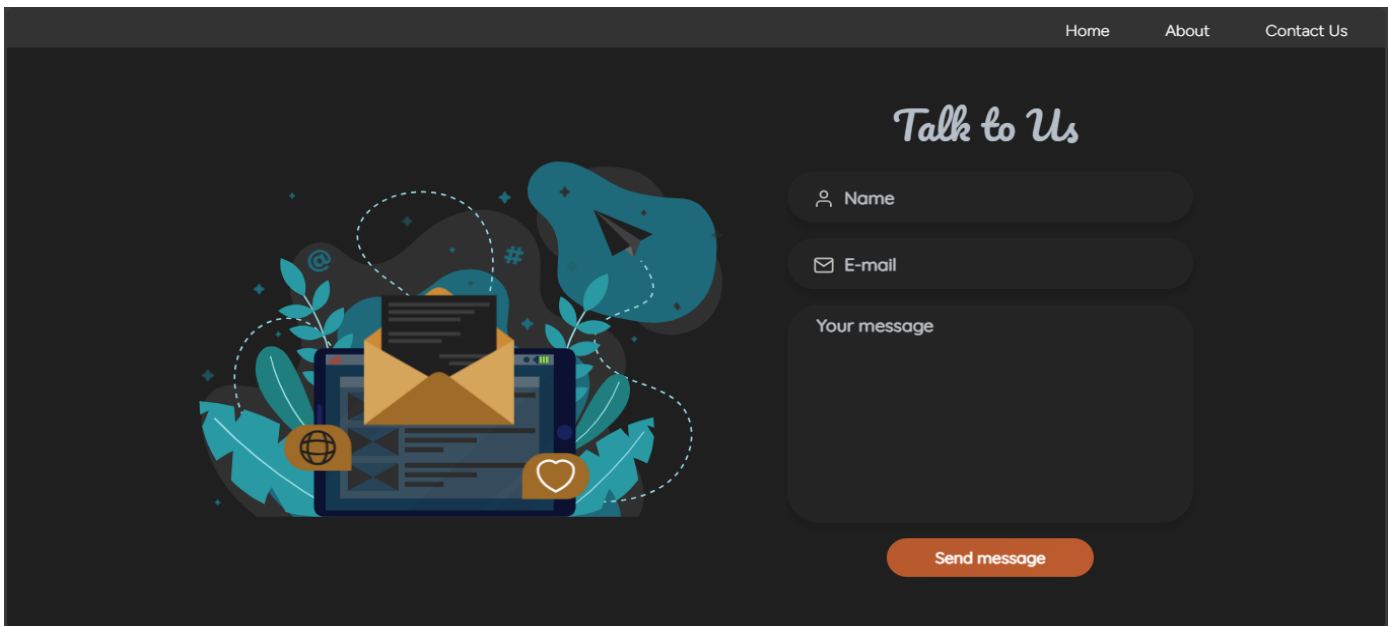
# WEB DEVELOPMENT

## LAB 2 ACTIVITY

RICHARD D. BILAN JR.

BSIT 3-C





This is Our project Looks Like. When users access the local host, the website will greet them as guests by default. The homepage is designed to provide a generic welcome message for all visitors who do not input specific details. However, if a user adds their name to the URL (e.g., /user/Richard), the website will personalize the greeting “Welcome, Richard!”. This feature allows the site to display a customized message based on the user's input.

## HTML (Structure)

### 1. Inside The Head Tags

```
<head>
  <meta charset="UTF-8">
  <title>About Us</title>
  <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap">
  <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap" >
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
  <link rel='stylesheet' href='https://fonts.googleapis.com/css2?family=Staatliches&display=swap'>
  <link rel='stylesheet' href='https://fonts.googleapis.com/css2?family=Bebas+Neue&family=Figtree:wght@300;600&display=swap'>
  <link rel="stylesheet" href="styles/styles_a.css">
</head>
<body>
```

- Since I was assigned to create this about webpage this is the First thing in my code. So, Inside the Head Tags has The ``<title>About Us</title>`` tag sets the title of the webpage that appears in the browser tab as "About Us." I search for Several Google Fonts and Font Awesome icons and imported it to use on the page, including Montserrat, which is a useful and clean font for body and heading text. I also found cool fonts like Bebas Neue, Staatliches, and Figtree and use it for headlines or branding purposes. the custom stylesheet is linked using the ``<link>`` tag, which connects to the external CSS file (`styles/styles_a.css`) that defines the specific styling for the page layout.

## 2. BODY

```
header>
  <nav class="list">|
    <ul>
      <li><a href="{{ url('/welcome') }}" style="text-decoration: none;">Home</a></li>
      <li><a href="{{ url('/about') }}" style="text-decoration: none;">About</a></li>
      <li><a href="{{ url('/contactus') }}" style="text-decoration: none;">Contact Us</a></li>
    </ul>
  </nav>
/header>
```

- The `<body>` tag contains all the visible content of the webpage. Inside the body, the `<header>` tag defines the navigation bar, which includes a list of links (Home, About, Contact Us). These links are styled as a simple list with inline links that direct users to routes like `/welcome`, `/about`, and `/contactus` using Laravel's `url()` helper.

```
<div class="section-header">
  <h2 class="header header_home">About Us</h2>
</div>

<div class="container">
  <div class="box">
    <div class="info-image">
      
    </div>
    <h2>RICHARD D. BILAN JR. </h2>
    <ul class="info-list">
      <li><strong>Education:</strong> <br>3rd Year BS Information Technology <br>College of Science <br>Bicol University</li>
      <li><strong>About Me:</strong> <br>I am an IT student with a strong curiosity about the tech field. While I am still exploring my<
    </ul>
  </div>
```

- So, here in the `<div class="section-header">` holds the **"About Us"** heading, which is centered and styled specifically with a larger font size thanks to its class. The `<div class="container">` wraps the main content area, using a grid layout to organize team member profiles into responsive columns. Each team member's profile is enclosed in a `<div class="box">`, which contains various elements such as the `<div class="info-image">` that holds the member's image, the `<h2>` tag that displays the member's name, and the `<ul class="info-list">`, which lists key details like their education and a brief bio.

## CSS (Style Sheet)

### 1. HTML, BODY

```
1  html {
2      width: 100vw;
3      box-sizing: border-box;
4      margin: 0;
5      padding: 0;
6  }
7
8  body {
9      overflow-x: hidden;
10     width: 100vw;
11     box-sizing: border-box;
12     margin: 0;
13     padding: 0;
14 }
```

- This is where the Style starts from its view for all the viewport Both the `<html>` and `<body>` elements are set to take up the full viewport width (100vw), verify or confirm that there is no horizontal scroll. Also , margins and padding are reset to zero to eliminate any default spacing added by the browser, creating a clean, edge-to-edge layout.

### 2. HEADER

```
16  header {
17      font-family: "Figtree", sans-serif;
18      justify-content: space-between;
19      align-items: center;
20      padding: 5px;
21      background-color: #333;
22      color: #e7e6e6;
23      margin-top: -8px;
24      width: 100vw;
25      box-sizing: border-box;
26  }
```

- The `<header>` is styled with the "Figtree" font and uses flexbox properties, such as `justify-content: space-between` and `align-items: center`, to align the navigation bar. The background is set to a dark color (`#333`), while the text is styled in a light color for contrast. The `<nav ul>` element displays the unordered list horizontally using flexbox, ensuring that the list items are evenly spaced. Additionally, the links have a hover effect that changes their color, providing interactive feedback for users.
- **section-header** The "About Us" title is centered, with a distinctive font size of 45px using the "Bebas Neue" font.

### 3. Container

```
5
6  .container {
7      display: grid;
8      grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
9      gap: 1px;
10     padding: 20px;
11 }
```

```
102 .box {
103     position: relative;
104     padding: 20px;
105     font-family: "Figtree", sans-serif;
106     box-sizing: border-box;
107     text-align: left;
108     border-radius: 5px;
109     transition: all 0.6s ease;
110     margin: 40px;
111     overflow: hidden;
112     display: flex;
113     flex-direction: column;
114     justify-content: space-between;
115     align-items: left;
116     min-height: 300px;
117     height: auto;
118     background-image: linear-gradient(45deg, #121336, #31312b);
119     background-size: 200% 100%;
120     animation: shine 1.5s infinite;
121 }
122
```

- The ``<div class="container">`` uses a grid layout to properly arrange profile boxes in a responsive grid that adjusts based on screen size, ensuring proper spacing for each box through the ``grid-template-columns`` property. Each ``<div class="box">`` features a gradient background with animated shining effects, along with padding, rounded corners, and a hover effect that alters its appearance by increasing the shadow and adding a glowing border. The ``box-shadow`` and transition effects enhance the visual interaction when the box is hovered over. Additionally, the profile images within the ``<div class="info-image">`` are styled to fit perfectly inside their boxes, with rounded corners and a consistent height of 250px, using the ``object-fit: cover`` property to ensure they are displayed correctly.

## 4. MEDIA

```
226 @media (max-width: 1000px) {
227     .box {
228         padding: 25px;
229     }
230 }
231
232 @media (max-width: 600px) {
233     .box {
234         margin: 10px;
235     }
236     .box h3 {
237         font-size: 20px;
238     }
239 }
240
241 @media (max-width: 768px) {
242     .section-header h2 {
243         font-size: 2rem;
244     }
245     .container {
246         grid-template-columns: 1fr;
247         align-items: center;
248         overflow-x: hidden;
249     }
250     .box {
251         text-align: center;
252     }
253 }
```

- In Media queries I used it to ensure that the design remains responsive across various screen sizes. On smaller screens, such as mobile devices, the layout adjusts to a single column, and font sizes or margins are modified to improve readability. The `@keyframes shine` animation creates a "shining" effect by moving a light gradient across the background of the boxes, adding a subtle and modern touch to the design.

## 5. ROUTES

```
19
20 Route::get('/user/{name?}', function ($name = null) {
21     $userName = preg_replace('/[^a-zA-Z]/', '', $name);
22
23     $userName = $userName ?: 'Guest';
24
25     session()->flash('user_name', $userName);
26     return redirect('/');
27 });
28
```

- This is the Routes of our Laravel so, This is how it works route manages the `/user/{name?}` URL pattern, where `{name?}` is an optional parameter. When a request is made to this route, the closure function processes the request. It first sanitizes the `\$name` parameter by removing any non-alphabetic characters using `preg\_replace`, ensuring that only letters are included in the `\$userName` variable. If the `\$name` is empty or null after this process, it defaults to `Guest`. The sanitized or default username is then stored in a session flash variable named `user\_name`, which will be available for the next request. Finally, the route redirects the user to the home page (`/`), where the flashed session data can be used to display the user name.