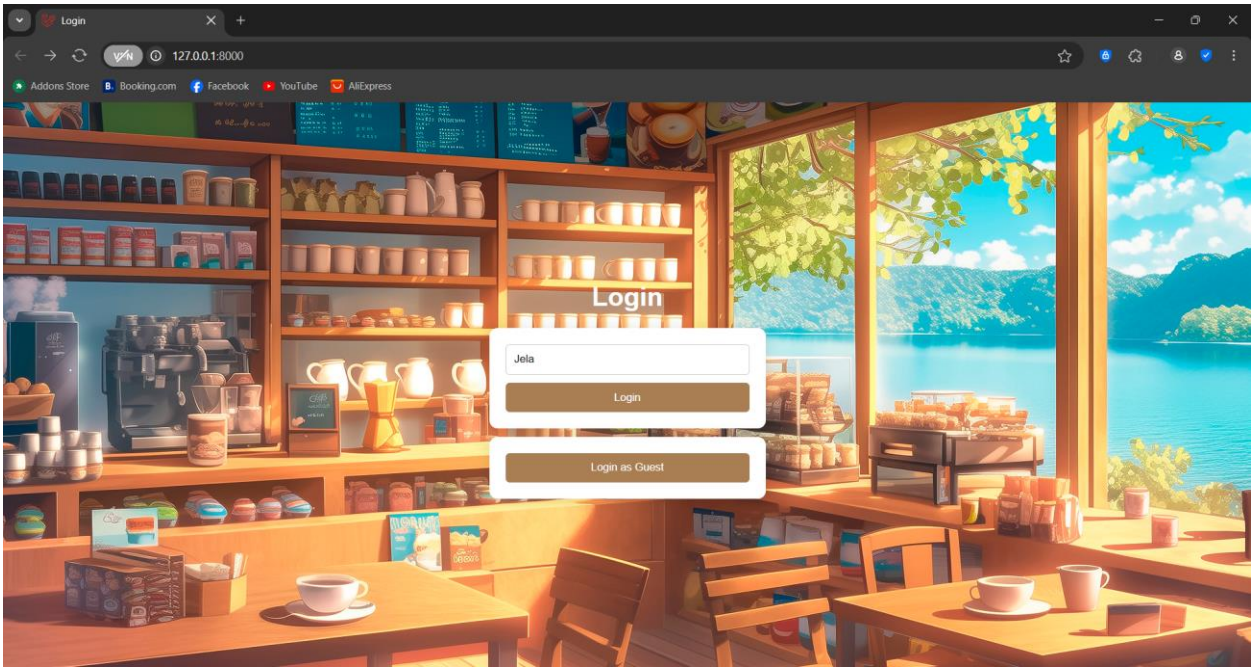
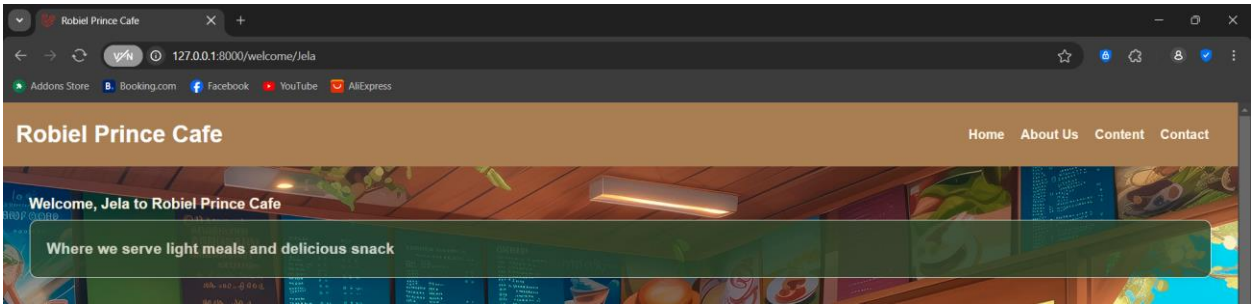
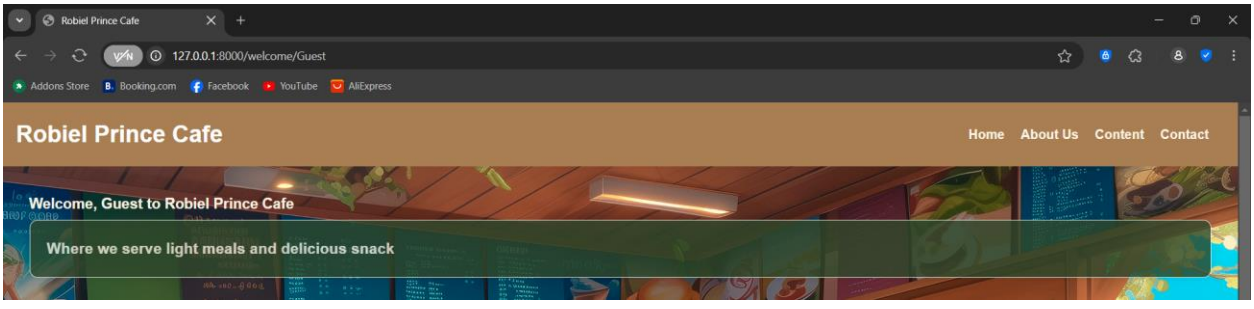
Jela Mae N. Gñilo **BSIT-3C**

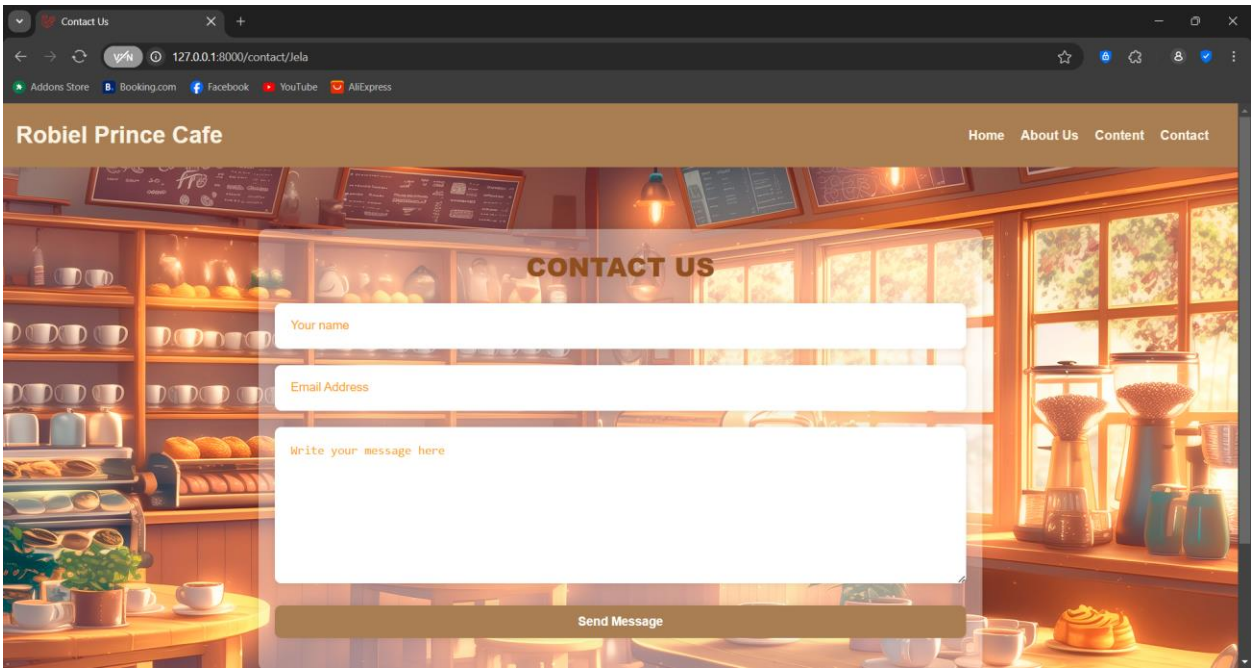
1. The first image shows the login page, and when the ‘Login’ button is clicked, the user is redirected to the welcome page. In the second image, ‘Welcome, Jela’ is displayed because a username was used for login. However, if no username is entered and the ‘Login as Guest’ button is clicked, the user will be redirected to the welcome page displaying ‘Welcome, Guest’.

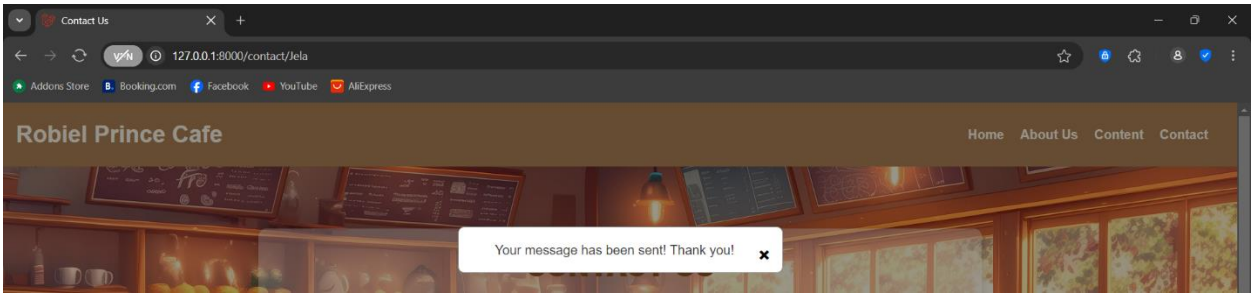


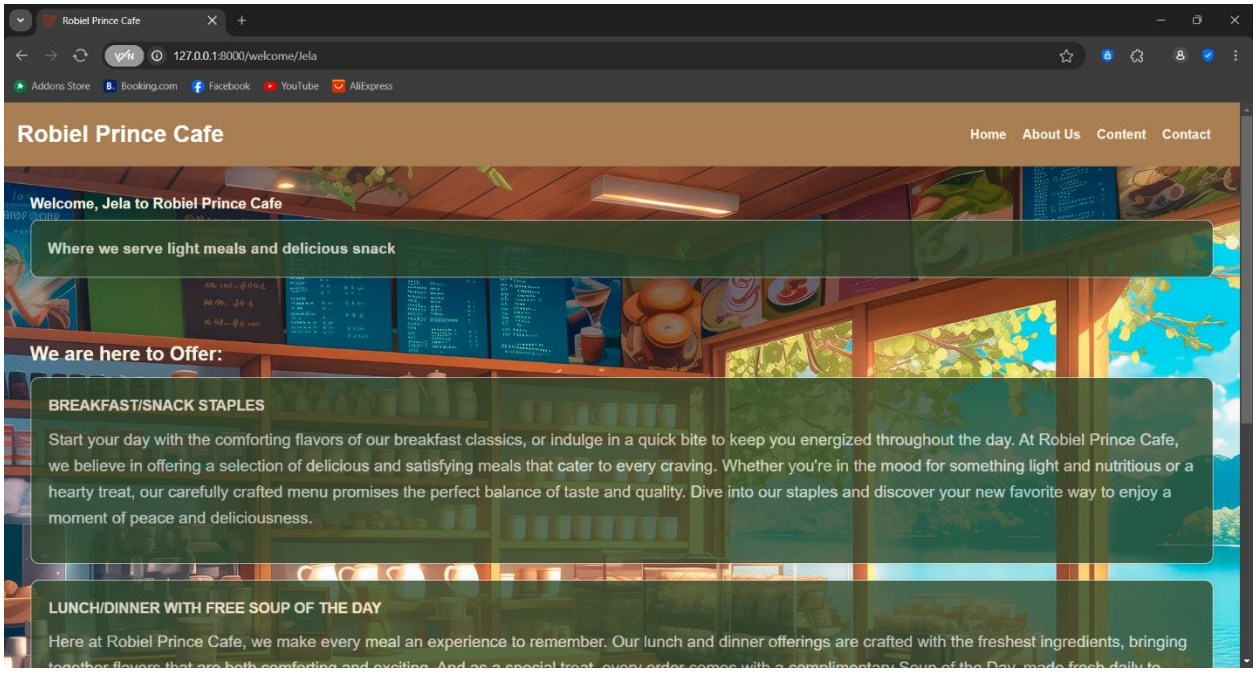




2. The ‘Contact Us’ page allows users to send their messages or concerns. After typing their message and clicking the ‘Send Message’ button, a pop-up will appear saying, ‘Your message has been sent! Thank you!’ After 2 seconds, the user will be redirected to the homepage.







3. **<a href="{{ route('welcome', ['username' => $username]) }}">:**
{{ route('welcome', ['username' => $username]) }}:

This Blade syntax generates a URL for the named route 'welcome', passing the username parameter.

route('welcome') generates the URL for the route named welcome.

The ['username' => $username] part dynamically inserts the username variable into the URL.

This will ensure that each navigation link is personalized with the current username.

If the username is Jela, the Home link might resolve to /welcome/Jela.

```
<ul class="nav-list">
    <li class="nav-item"><a href="{{ route('welcome', ['username' => $username]) }}">Home</a></li>
    <li class="nav-item"><a href="{{ route('about', ['username' => $username]) }}">About Us</a></li>
    <li class="nav-item"><a href="{{ route('content', ['username' => $username]) }}">Content</a></li>
    <li class="nav-item"><a href="{{ route('contact', ['username' => $username]) }}">Contact</a></li>
</ul>
```

4. The Route::post method defines a route that listens for POST requests to the /welcome URL. This means that when a user submits the login form, this route will be triggered.

   The callback function that follows (function (\Illuminate\Http\Request $request)) is responsible for handling the incoming request.

```
//handles the log in form and username validation
Route::post('/welcome', function (\Illuminate\Http\Request $request) {
```

5. validate: This method validates the incoming request. It checks the username field for specific rules:

   required: The username field must be filled out (it cannot be empty).

   alpha: The username can only contain alphabetic characters (letters a-z or A-Z). No numbers, spaces, or special characters are allowed.

   The second argument in the validate method is an array that defines custom error messages for each validation rule.

   username.alpha: If the username contains any characters other than alphabetic ones, this error message will be displayed: "The username should only contain alphabetic characters."

   username.required: If the user submits the form without entering a username, this error message will be displayed: "The username field is required."

```
// Validate that the username only contains alphabetic characters (a-z, A-Z)
$request->validate([
    'username' => 'required|alpha'
], [
    'username.alpha' => 'The username should only contain alphabetic characters.',
    'username.required' => 'The username field is required.'
]);
```

6. This line retrieves the value of the username field from the form. If the user doesn't provide a username (though in this case, it's validated to be required), it defaults to 'Guest'.
   After validating the username, the user is redirected to the /welcome/{username} route. The {username} part will be replaced with the actual username the user entered (or 'Guest' if no username is provided).

```
    $username = $request->input('username', 'Guest');
    return redirect("/welcome/$username");
});
```

7. **URL**: / (root URL)
   Method: GET
   Action: Returns the login view when someone visits the root URL (usually the homepage).
   This is the route for the login page where users will enter their username. The name('login') part assigns the route a name, allowing you to easily reference it elsewhere in your code (e.g., in form actions or redirects).

```
//route for the login page at the root URL "/"
Route::get('/', function () {
    return view('login');
})->name('login');
```

8. **URL**: /welcome/{username} (e.g., /welcome/Jela)
   Method: GET
   Action: Displays the welcome view, passing the dynamic username as data to the view. The view can use this username to display a personalized welcome message.
   Once the user is logged in, they will be redirected here. The {username} part of the URL is dynamic, meaning it will display the username entered by the user.

```php
//route for the welcome page with dynamic username, and view the welcome message with username
Route::get('/welcome/{username}', function ($username) {
    return view('welcome', ['username' => $username]);
})->name('welcome');
```

9. Each route handles different pages (login, welcome, about, content, and contact).

   They all pass a username parameter dynamically through the URL, which is then passed to the corresponding views.

   This setup allows for personalized pages where the username is dynamically displayed based on the user's login session or input.

   Each route is named using the name() method, making it easier to generate URLs or redirect users programmatically.

```php
//route for the login page at the root URL "/"
Route::get('/', function () {
    return view('login');
})->name('login');

//route for the welcome page with dynamic username, and view the welcome message with username
Route::get('/welcome/{username}', function ($username) {
    return view('welcome', ['username' => $username]);
})->name('welcome');

//route for the about page with dynamic username
Route::get('/about/{username}', function ($username) {
    return view('about', ['username' => $username]);
})->name('about');

//route for the content page with dynamic username
Route::get('/content/{username}', function ($username) {
    return view('content', ['username' => $username]);
})->name('content');

//route for the contact page with dynamic username
Route::get('/contact/{username}', function ($username) {
    return view('contact', ['username' => $username]);
})->name('contact');
```