**Eurtech- Documentation**  
  
This documentation covers the Header component, IT Solution component, Service component, Testimonials component, and Footer component.

**Steps to Create the Page**

This section outlines the detailed steps followed to create the page, including the creation of components, templates, models, controllers, and the integration with Sitecore.

**1. Create the Page Component by Component**

* **Objective:** Break down the page into individual components to manage the development process effectively.

**2. Start with the Three Main Components**

* **Components:** Begin with the primary components of the page, which typically include two listing components followed by the header and footer.
* **Approach:**
  + Identify the main components required for the page.
  + Develop these components first as they form the backbone of the page structure.

**3. Create Templates**

* **Templates Creation:**
  + Develop templates for each component.
  + Create templates for subitems in listing components to ensure a structured approach.
* **Implementation:**
  + Define the fields required for each component in the template.
  + Ensure consistency across templates for similar components.

**4. Create Items Using Templates**

* **Items and Subitems:**
  + Use the templates to create items for each component.
  + For listing components, create subitems to populate the lists dynamically.
* **Sitecore Interface:**
  + Navigate to the Sitecore content tree.
  + Create items under the appropriate parent item, following the structure defined by the templates.

**5. Develop Model, Controller Action Method, and View**

* **Model:**
  + Create a model class for each component to represent the data structure.
* **Controller:**
  + Define a controller action method to fetch and process the data.
  + Ensure the controller action returns the model data to the view.
* **View:**
  + Create a view for each component to render the data on the frontend.
  + Use Razor syntax to bind model data to the view.

**6. Fetch Sitecore HTML Items and Implement Styling**

* **Fetch Data:**
  + Use Sitecore's API to fetch HTML items.
  + Bind these items to the model in the controller action method.
* **Styling:**
  + Implement CSS styles separately in a dedicated folder.
  + Ensure styles are modular and reusable across components.
  + Use classes and IDs to target specific elements within the components.

**7. Create Controller Rendering for the Component in Sitecore**

* **Controller Rendering:**
  + Define a controller rendering for each component in Sitecore.
  + Specify the controller, controller action, and view to be used for rendering the component.

**8. Add Rendering of All Components to the Home Page in Presentation Details**

* **Presentation Details:**
  + Navigate to the Home page item in Sitecore.
  + Edit the presentation details to add renderings for each component.
* **Data Source and Placeholders:**
  + Provide the data source for each component to bind the appropriate content.
  + Specify the placeholders where each component should be rendered on the page.

**9. Make Necessary Components Cacheable**

* **Caching:**
  + Identify components that can be cached to improve performance.
  + Configure caching settings in Sitecore to cache the output of these components.
* **Implementation:**
  + Set caching options such as duration, vary by parameters, and cacheable checkboxes.
  + Ensure cache settings do not interfere with dynamic content updates.

Below are the components in detail:

**Header Component**

**Description:** The header component consists of an external structure template with three primary items: a logo, a navigation bar (navbar), and a "Get Appointment" button.

**Templates:**

1. **Header External Structure Template:**
   * Logo
   * Navbar (Multilist)
   * Get Appointment Button
2. **Navbar Template:**
   * This is a multilist component containing six navigation items.
3. Controller Code:

A screen shot of a computer program

Description automatically generated

1. Model Code:

A screen shot of a computer program

Description automatically generated

**ITSolution Component**

**Description:** The ITSolution component serves as a dynamic hero banner section for a webpage. It displays key information such as background imagery, headings, subtext, and buttons, with an emphasis on visual presentation and usability.

**Templates:**

1. **ITSolution Template:**
   * + SubText (Type: Single-Line Text)
     + Heading (Type: Single-Line Text)
     + SubHeading (Type: Single-Line Text)
     + Button1 (Type: General Link)
     + IntroLink (Type: General Link)
     + BgImg (Type: Image)
2. Controller Code:

A screen shot of a computer program

Description automatically generated

1. Model Code:

A screen shot of a computer program

Description automatically generated

**Service Component**

**Description:** The service component uses three items: a preheader, title, and a serviceCard item (a multilist component). Additionally, a separate template for serviceItems was created.

**Templates:**

1. **Service Component Template:**
   * Preheader
   * Title
   * ServiceCard (Multilist)
2. **ServiceItem Template:**
   * Image
   * Title
   * Description
   * Link Text
3. Controller Code:

A computer screen shot of a program code

Description automatically generated

1. Model Code:
2. A screen shot of a computer program

   Description automatically generated

**Testimonials Component**

**Description:** The testimonials component is a listing component with three items: a preheader, a title, and a multilist component for TestimonialCard. An internal template for the testimonial card was created with four items.

**Templates:**

1. **Testimonials Component Template:**
   * Preheader
   * Title
   * TestimonialCard (Multilist)
2. **TestimonialCard Template:**
   * Quote
   * ProfileImage
   * Author
   * Designation
3. Controller Code:

A computer screen shot of text

Description automatically generated

1. Model Code:

A screen shot of a computer program

Description automatically generated

**Footer Component**

**Description:** The footer component is a listing component with an external structure template. It includes a multilist for four different columns, a droplink for the footer navbar, and a footer text.

**Templates:**

1. **Footer External Structure Template:**
   * Columns (Multilist)
   * Footer Navbar (Droplink)
   * Footer Text
2. **Single Column Template:**
   * Title
   * Ten Text Items
3. **Footer Navbar Template:**
   * Logo
   * Three Text Items for Navigation
4. Controller Code: A computer screen shot of a program

   Description automatically generated
5. Model Code:

A screen shot of a computer program

Description automatically generated