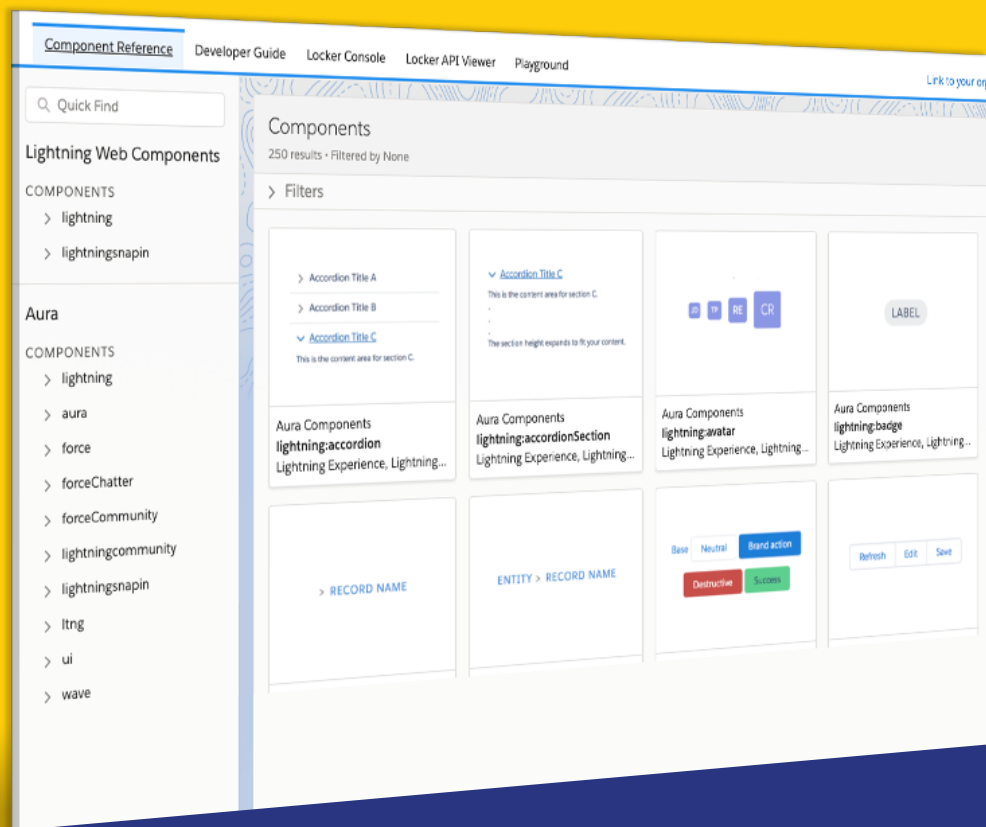


Customise TargetRecruit and Bring your Ideas to Life



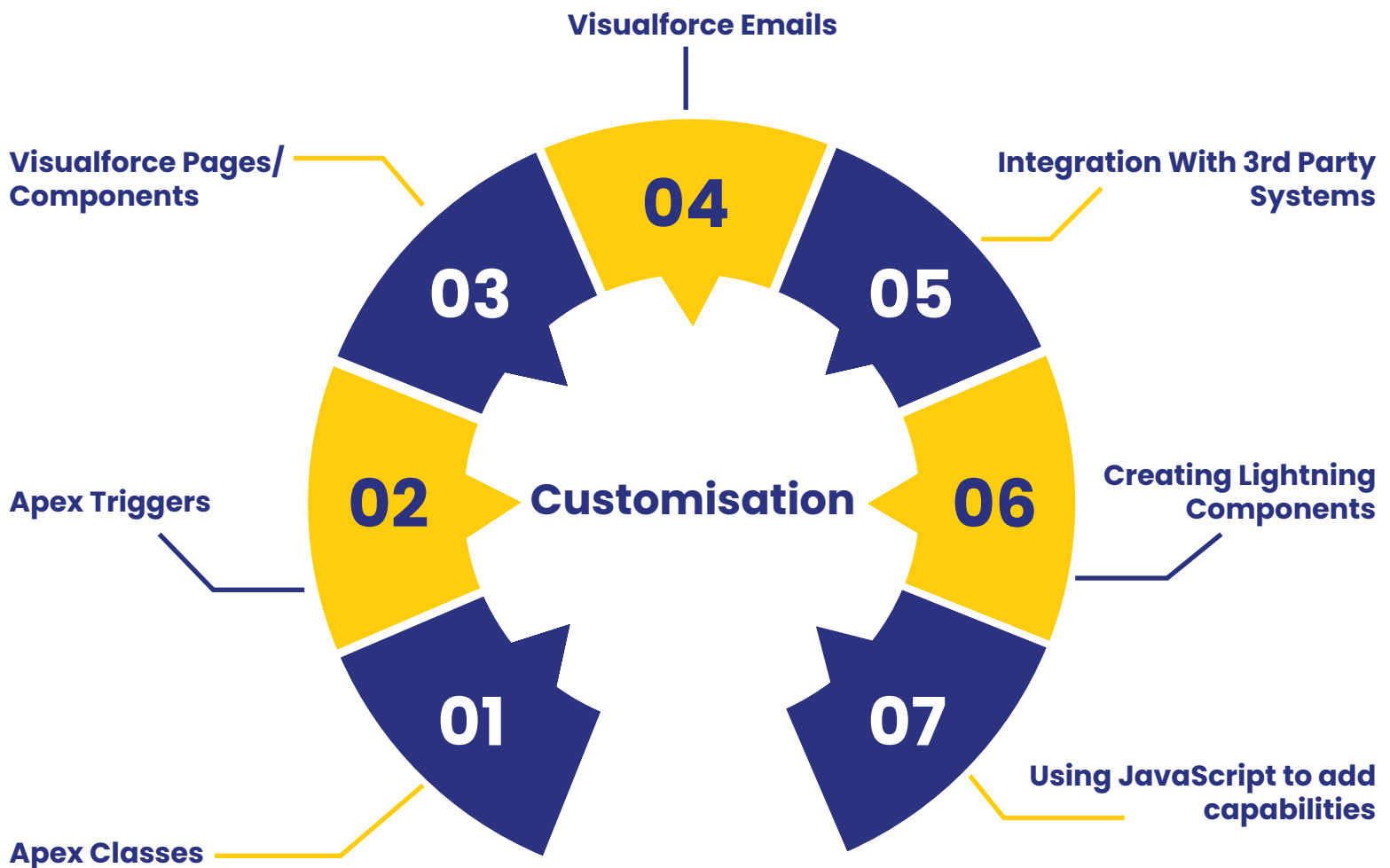
Built to Scale

Flexibility is the hallmark of any true enterprise software solution. As your business scales, you need the ability to add your own features and functionality quickly and cost effectively. TargetRecruit includes the embedded Salesforce Platform licenses that gives you the option to enhance existing functionality or add completely new features – all within our solution.

Choose Your Team

The Salesforce Platform empowers those who are not developers to create and customise solutions that may otherwise require coding – this applies to TargetRecruit as well. Salesforce offers an excellent learning experience platform called Trailhead that is designed for beginners as well as advanced users. For more sophisticated development, you have various options available:

- Engage our Professional Services to provide customised solutions inexpensively.
- Build an in-house team or use contractors, of which there are over 17,000 globally.
- Hire a Salesforce System Integrator (SI), of which there are over 1,000 globally.



Custom Objects

Create custom objects with clicks not code using simple drag-and-drop technology

Custom Objects allow you to track and store data within TargetRecruit that can be related to an existing Object – a record type such as a client, person, job or placement. It is often useful to have the ability to add additional fields to an existing record to store information that is unique to your business, but imagine the potential to create whole new tables of data where needed.

Using Custom Objects allows the unlimited build-out of the standard application to meet whatever needs your business may have, while preserving the standard data structures and relationships. In this way you can build a whole additional set of functionalities that integrates seamlessly with your existing data and workflows without the need for a separate system.

Lightning Web Components (LWC)

Build modern apps for desktop and mobile

The Salesforce Lightning Web Component framework was introduced in 2014 and is a modern User Interface that develops single page web applications for desktop and mobile devices. It is a faster and smarter way of developing and deploying innovative applications with quality UI on the Salesforce platform. Accelerating development and app performances, it builds dynamic, responsive user interfaces for any device. Using JavaScript on the client-side and Apex on the server side, it offers exciting tools to developers allowing them to deploy new apps and build reusable building blocks, that are custom components, for seamless customisations.

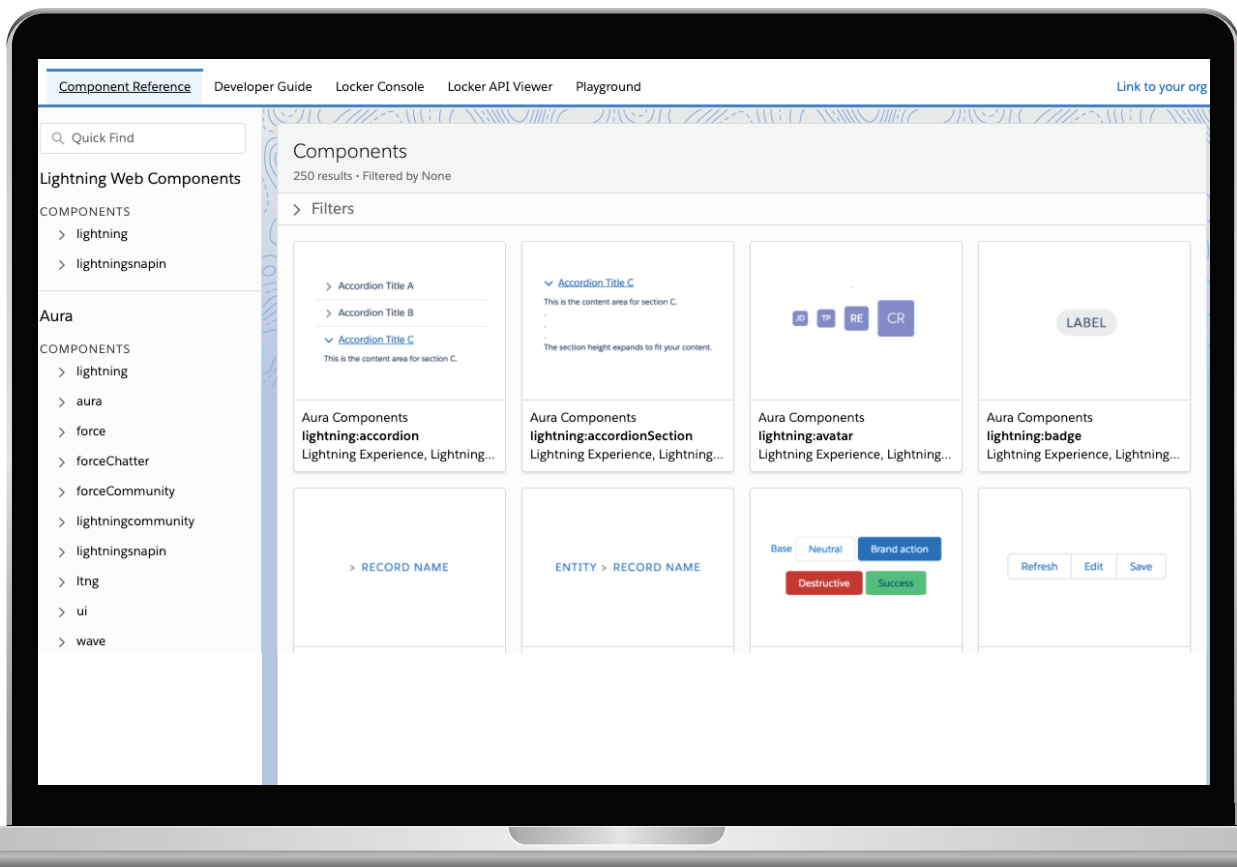
You can build Lightning components using two programming models – the Lightning Web Components and the original Aura Components. Lightning Web Components are custom HTML elements that are built using HTML and JavaScript. Lightning web components and Aura components may coexist and interact on a page.

Lightning Web Components framework brings power and benefits of Web Components while being built on modern web standards. Lightning Web Components are comprised of three key pieces:

Base Lightning Components – a set of 80+ UI components built as custom elements.

The Lightning Data Service– provides declarative access to Salesforce data and metadata, data caching, and data synchronisation.

The User Interface API– the underlying service that makes Base Lightning Components and the Lightning Data Service metadata-aware.



Extend with code using Visualforce

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that are consumed within the TargetRecruit app. The Visualforce framework includes a tag-based mark-up language, similar to HTML, and a set of server-side “standard controllers” that make basic database operations, such as queries and saves, very simple to perform.

Your Visualforce toolkit includes:

- Visualforce Page– A markup language like HTML is used for designing the layout of pages.
- Custom Controller– Apex code is used to handle backend processing (server-side operations) implemented at the VF page.
- JavaScript– JavaScript is optional and helps in managing client-side processing and can be coupled with CSS for a revamp of the interface.
- Apex Extensions– Logical operations can be performed using Apex extensions which are not available in standard controllers. However, this is optional too.

With Visualforce you can:

- Create custom user interfaces that easily leverage standard Salesforce styles
- Create custom user interfaces that completely replace the standard Salesforce styles
- Build wizards and other navigation patterns that use data-specific rules for optimal, efficient application interaction

Visualforce comes with a rich component library that allows you to quickly build pages without having to create a lot of functionality yourself. In the Visualforce markup language, each tag corresponds to a coarse or fine-grained component, such as a section of a page, a related list, or a field. The components can either be controlled by the same logic that is used in standard Salesforce pages, or developers can associate their own logic with a custom controller or controller extension written in Apex.

Visualforce Pages

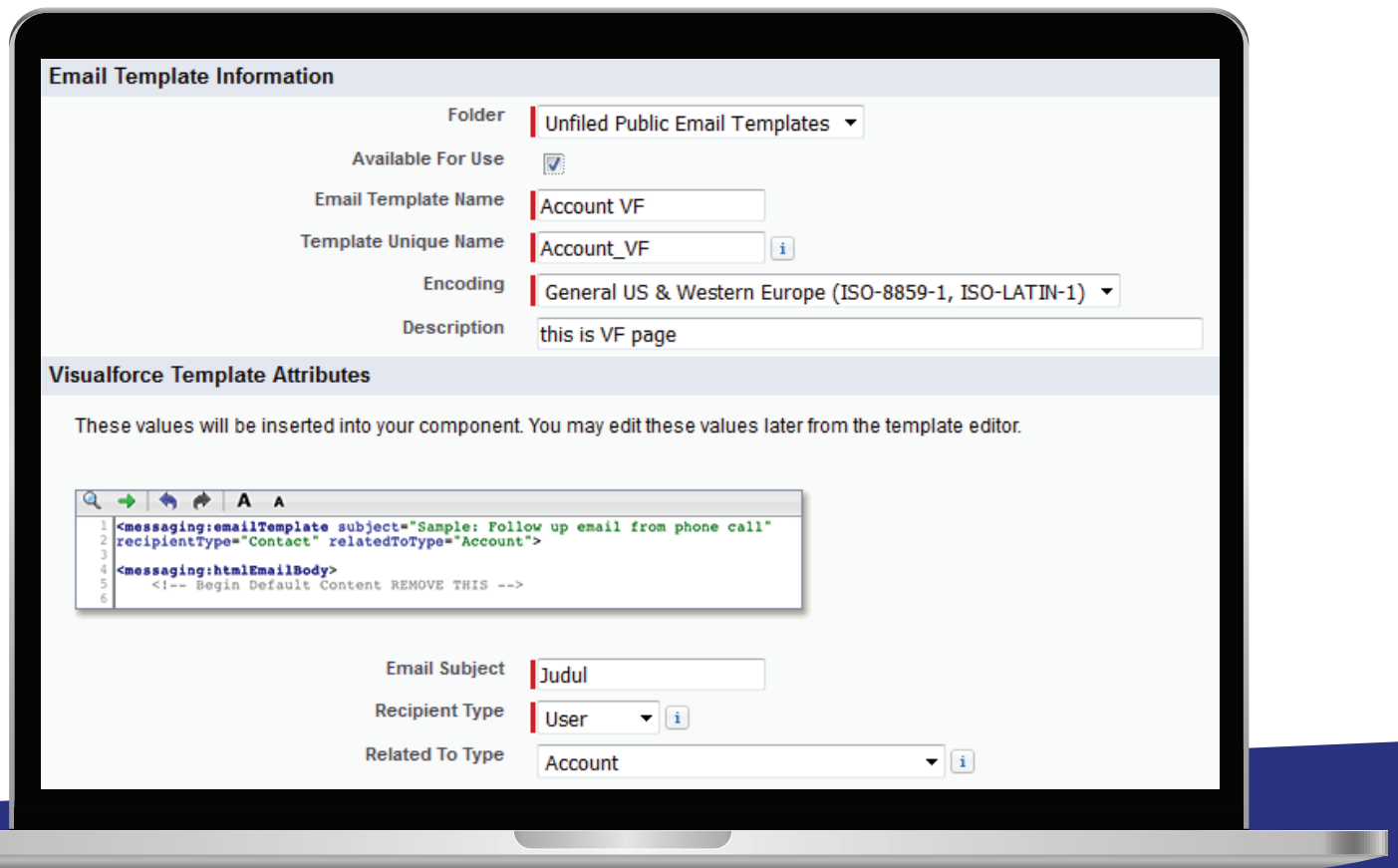
Visualforce pages are the top-level container for custom apps built with Visualforce. Create Visualforce pages by adding Visualforce components (standard or custom), static HTML mark-up, and CSS styles and JavaScript to the page.

Visualforce Components

Visualforce components are small, reusable pieces of functionality—think widgets, panels, user interface elements, that kind of thing—that you use in Visualforce page markup. You can use standard Visualforce components, and create your own custom components.

Static Resources

Static resources allow you to upload content that you can reference in a Visualforce page, including archives (such as .zip and .jar files), images, style sheets, JavaScript, and other files.



Visualforce or Lightning? Which to choose

The Visualforce framework provides a robust set of tags that are resolved at the server-side and that work alongside standard or custom controllers to make database and other operations simple to implement. This is a page-centric web application model. It's great for basic functionality, but it's challenging to deliver the new, more dynamic experience that users expect. Fundamentally, this is because it relies on the server to generate a new page every time you interact with the application.

Lightning components are part of the new Salesforce user interface framework for developing dynamic web applications for desktop and mobile devices. They use JavaScript at the client-side and Apex at the server-side to provide the data and business logic. To deliver a more interactive experience, you need help from JavaScript on the client-side. In this new app-centric model, JavaScript is used to create, modify, transform, and animate the user interface rather than completely replacing it a page at a time. This model is exciting, interactive, and fluid. Both the page-centric and app-centric models have their own advantages and both are here to stay. Combining the models lets applications deliver the right type of experience for the right use case.

	Visualforce	Lightning
UI Generation	Server-side	Client-side
Advantages	Tried and true model	Enables highly interactive user experiences
	Easy to implement	Aligns with Salesforce user interface strategy
	Naturally splits large applications into small, manageable pages	Build on metadata from the foundation, providing an integrated developer experience
Disadvantages	Limited interactivity	Higher learning curve
	Higher latency	Higher complexity than Visualforce – you're building an app, not a page
		Some features are not supported
		There are limited out-of-the-box components

Use a Sandbox Prior to Deployment

Each change you make to a complex system can cause any array of unpredictable consequences that cannot be rolled back. To avoid disrupting your operations, you need to understand how different objects and features work. As it is not always possible to foresee each small detail, it's much safer to test your customisations in a sandbox environment that does not impact your production system in any way.

There are four sandbox types in Salesforce:

Salesforce Sandbox Types

Developer Sandbox

Used for creating new customisations and testing them with fresh test data in a clean system - this sandbox is created without any historical data.

Developer Pro Sandbox

Used for creating new customisations and testing them with larger sets of fresh test data than in the Developer Sandbox. The system is also isolated and clean.

Partial Copy Sandbox

Used as a testing environment with a set of sample data, to check how a new customisation can affect the system.

Full Sandbox

Used to carry out performance testing, load testing and staging. It contains a full copy of the system so it's used for final testing before going live.

The Salesforce Platform Puts Development Power in Your Hands

No one understands your business like you do, and with the power to create custom functionality in your hands, you will be able to turn that expertise into real, personalised solutions, and to do so more quickly than with any other platform.

Contact us to learn more about TargetRecruit's customisation capabilities, powered by Salesforce.