

# Simun - Supplementary

The public repo can be found on [GitHub](#).

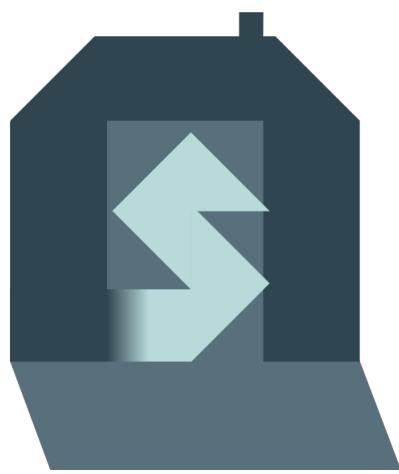
## Table Rankings

Category	Rank	Where to Find/Why
HTML	A	Pages are delivered using pug (jade) as templates, while following xHTML rules. There are 7 different page templates.
CSS	A	CSS is compiled using node-sass, allowing nested CSS styles. A lot of CSS is used, including CSS keyframe animations. There are no style tags or style attributes in the HTML.
JS	A	Client side javascript in Simun is used for changing appearance of objects (e.g. resizing outbox image selector size or opening snippet-selected divider in the inbox), requesting data from the server (see the tools.js script in model), or updating the page depending on the result of that variables request (see setting snippets active in the inbox).
PNG	A	Gimp was used for creating and editing images. Many different tools were used for the original site, which attempted to have a pseudo realistic house as the setting, with mail arriving in the postbox. See old site design (p2) for reference. Some of these tools include magnetic lasso, handling layers for different objects, airbrushing (and original artwork, although it is very ugly), magic wand for selecting objects, then tidying it up by erasing edges and using smudge.
SVG	A	Inkscape was used for creating SVGs. All designs are custom made, and work well on the web page as they use block colours and paths. Originally methods such as patterns and gradients were used, such as in the original simun logo, but stopped being used as they did not suit the final design theme. In the final design theme some techniques I used were swatches for colours, layers, lots of grid snapping, bezier curves for curved shadow effects and path editing for constructing unusual shapes. I experimented with lots of other tools, for example applying filters (p3).
Server	A	The server is hosted live on <a href="#">simun.co.uk</a> on a virtual machine provided my Microsoft Azure. HTTPS is used, which has prior dependencies, and pages are loaded differently depending on browser (mainly due to css though). There is unit testing using modules mocha and chai, which tests that server and database logic and authentication works as expected.

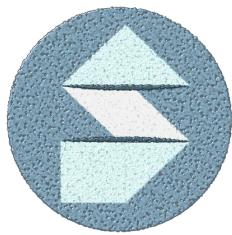
Category	Rank	Where to Find/Why
<b>Database</b>	<b>A</b>	The database uses SQLite3 to interface with the stored .db file. There is a significant amount of data to handle, and effort is made to reduce this, for example by storing images in imgur and linking their url rather than storing them directly in the database. The full database schema can be seen in the main report.
<b>Dynamic Pages</b>	<b>A</b>	Pages are delivered populated with content, and can then retrieve new content from the server using client side scripts. Node.js and express.js are used extensively for delivering pages, managing cookies and negotiating content.



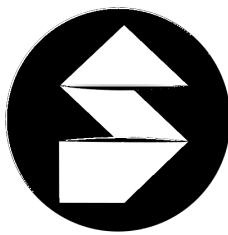
Original Simun Appearance



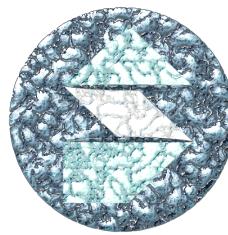
Original Simun Logo



Bubbly



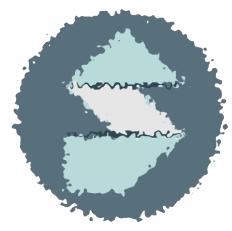
Bump Engraving



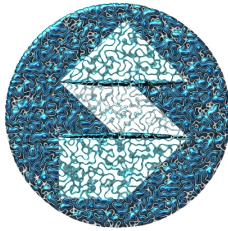
Cracked Lava



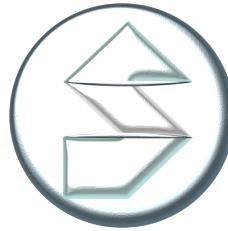
Glowing Bubbles



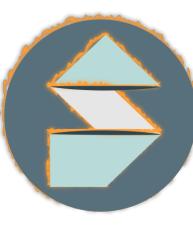
Lapping



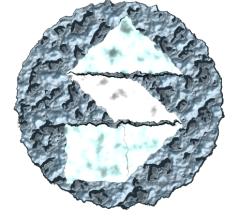
Lizard Skin



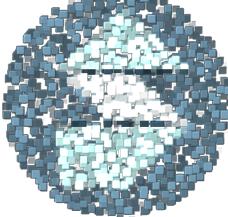
Metallic



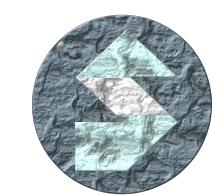
Fire Protrusion



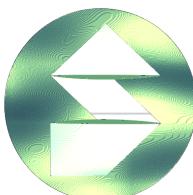
Riddled



Scatter Cubes



Thick Acrylic



Tinfoil

## Inkscape Filters Experimentation