

## Art Gallery Website

### MONGODB/EXPRESS BACKEND

Items	Instances
Each document in your MongoDB (embedded or separate), with at least <b>two keys</b> (excluding IDs).	<ul style="list-style-type: none"><li>• Each document has multiple keys e.g. post-date, poster name, image link, art title, art type, art subject, art description, statistics (like count and review count) and reviews etc.</li></ul>
Each <b>1:M relationship</b> in your MongoDB, embedded or separate.	<ul style="list-style-type: none"><li>• One art post can have multiple reviews</li><li>• One art type (e.g. digital, traditional) can have many art posts</li><li>• One poster can create multiple art posts</li></ul>
Each <b>M:N relationship</b> in your MongoDB, embedded or separate	<ul style="list-style-type: none"><li>• One art post can have multiple subjects (e.g. animal, nature, people)</li><li>• One subject can have many art posts</li></ul>
Each updating of a sub-document in a document (singular, not array)	<ul style="list-style-type: none"><li>• Updating review count for each art</li><li>• Updating like count for each art</li></ul>
Each updating of a sub-document in an array inside a document	<ul style="list-style-type: none"><li>• Allow users to create, update, and delete reviews.</li><li>• Review dates are automatically updated upon creation and update.</li><li>• Reviewer's name can be created and updated.</li></ul>
Each demonstration of a MongoDB <i>find()</i> technique (eg. find with \$gte, find with \$or etc.)	<ul style="list-style-type: none"><li>• \$and – filter for a combination of art type and art subject</li><li>• \$nin – return all the art post except for the current one</li><li>• \$all – return all the results that contains the different art subjects</li></ul>

	<ul style="list-style-type: none"> <li>• \$ or – return all the results that contains either the name of the poster or art name</li> </ul>
Each set of <b>CRUD</b> endpoints your app	<ul style="list-style-type: none"> <li>• CRUD endpoints for art posts</li> <li>• CRUD endpoints for reviews</li> </ul>
<b>Bonus:</b> Every feature not listed above	-

## REACT FRONTEND

Items	Instances
Basic Form processing for each of those component: <ul style="list-style-type: none"> <li>• Text input</li> <li>• Text area</li> <li>• Radio button</li> <li>• Dropdown select single</li> </ul>	<ul style="list-style-type: none"> <li>• Text input → ArtForm.js</li> <li>• Text area → ArtForm.js, ArtInfo.js</li> <li>• Radio button → ArtForm.js</li> <li>• Dropdown select single → App.js</li> </ul>
Advanced Form processing for each of those component: <ul style="list-style-type: none"> <li>• Multi-select checkbox</li> <li>• Multi-select dropdown</li> </ul>	<ul style="list-style-type: none"> <li>• Multi-select checkbox → App.js, FilterOptions.js</li> </ul>
Changing of CSS using React, applicable for: <ul style="list-style-type: none"> <li>• color</li> <li>• font-style</li> <li>• display</li> <li>• background-color</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• All buttons changes background colour on hover</li> <li>• Positioning of art type tags</li> <li>• Model modal popup using css display</li> <li>• Background-image is set using React inline styling for hero and each art post</li> <li>• Font-style are customized using React</li> <li>• Font colours are set using React</li> </ul>

Conditional rendering	<ul style="list-style-type: none"> <li>• Rendering of display using state variables to close popups or switch to a different “page”</li> <li>• Conditional rendering of preview image in ArtForm.js</li> </ul>
Advanced interaction/navigation techniques via React (modal pop-up, single page application, toastr etc.)	<ul style="list-style-type: none"> <li>• Modal pop-up</li> <li>• Single page application</li> </ul>
Use of state variables and binding those state variables to display or behaviour in a component <b>(excluding form processing)</b>	<ul style="list-style-type: none"> <li>• Display review count</li> <li>• Display like count</li> <li>• Display reviews</li> <li>• Display the different “pages”</li> </ul>
Each instance of list rendering (outside of CRUD context)	<ul style="list-style-type: none"> <li>• Render tags for art subject for each art post in App.js</li> </ul>
Each set of CRUD	<ul style="list-style-type: none"> <li>• CRUD for art posts</li> <li>• CRUD for reviews</li> </ul>
<b>Bonus:</b> Every feature not listed above	-
<b>Bonus:</b> Use of multiple components and props to influence the behavior of those components	<ul style="list-style-type: none"> <li>• Class components: <ul style="list-style-type: none"> <li>○ App.js, ArtInfo.js, CreateArtPage.js, EditArtPage.js, EditReviewPage.js, Filteroptions.js</li> </ul> </li> <li>• Function component: <ul style="list-style-type: none"> <li>○ ArtForm.js</li> </ul> </li> </ul>