

	<b>Learning Outcome</b>	<b>Fail</b> 0-39%	<b>Third</b> 40-49%	<b>Lower Second</b> 50-59%	<b>Upper Second</b> 60-69%	<b>First</b> 70-100%
<b>API Design</b> 20% ILO 1	Develop a secure, open-standards-based API to support server-client communication.	Failed attempt to implement an API.	Working API but not REST-compliant. Images stored in static directory. Some routes are not used by the front end.	API is Fremont <b>stage 2</b> compliant with correct use of methods, collections resources and status codes. Files in public directory.	API that complies with Fremont <b>stage 3</b> with correct implementation of JSON:API. Files returned as Data URLs.	API that complies with the Fremont REST Maturity Model <b>stage 5</b> , with full JSON:API with responses including complete HATEOAS.
<b>Security</b> 20% ILO 1		The API has no security or a failed attempt to implement security.	Some API routes prevent access without a valid Basic Auth token.	All API routes prevent access without a valid Basic Auth token.	All API routes only allow access with a valid Basic Auth token and to the right users.	Role-based authorization correctly implemented. CORS restrictions added. API uses OAUTH or JWT.
<b>API Calls</b> 10% ILO 2	Create modern web content involving asynchronous data retrieval, client-side DOM manipulation, standards adherence and user-user interaction.	Website does not implement fetch requests to the API or third-party front-end frameworks used.	Simple fetch requests implemented which return correctly structured JSON data.	Every page fetches live data from the API and uses this to display content to the user.	API calls used a range of http methods appropriately and the front-end handles http errors and issues.	API implements pagination, searching, filtering and partial responses which is used to display data to the user.
<b>DOM</b> 10% ILO 2		No ES6 JS DOM manipulation in the front-end or third-party front-end frameworks used.	Simple DOM manipulation used to change page content responding to route.	DOM manipulation based on route and content loaded via API calls. Some use of html templates to store DOM fragments.	A range of advanced DOM manipulation including interactive dynamic page events with html templates for all DOM fragments.	A range of advanced DOM manipulation including dynamic page events and custom elements to simplify DOM manipulation.
<b>Data</b> 10% ILO 3	Manage data persistence across both server and client web-based solutions.	Some page content is stored as static HTML in the SPA or stored as JSON strings in the API or no tables added to the db.	MySQL database is configured in the cloud and used to store user-generated content.	MySQL database in the cloud used to store all page content. Some attempt at database normalisation.	Database fully normalised to 3NF using the most appropriate data types of the optimal size.	All page content stored in a fully 3NF normalised MySQL database with a complete ORM model implemented.
<b>Functionality</b> 30% ILO 4	Design and implement an API and client based on given, non-trivial requirements using a range of appropriate developer tools.	Wrong topic, feature 1 not complete or no testing accounts or not using template or Screenshot does not match live website.	Feature 1 <b>fully</b> implemented and demonstrated. The screencast shows <i>real</i> website functionality.	Features 1 and 2 <b>functionality fully</b> implemented and demonstrated.	Features 1, 2 and 3 <b>fully</b> implemented and demonstrated.	Features 1, 2, 3 and 4 <b>fully</b> implemented and demonstrated.

