|  |
| --- |
| ATWD II PROJECT PROPOSAL WS269879 |
| I am proposing an online web application which will allow an authenticated user to browser various data from Dungeons & Dragons 5th Edition using the data provided from a third party publicly accessible API <http://www.dnd5eapi.co/> . The project will be called D&D 5e Explorer (DD5e explorer).  This application will contain both a frontend client and backend API. The frontend application which will allow the ability to browse the dnd5eapi and my backend. My backend API will allow for secure access to my database to ensure safe communication between clients and the database.  While the application will be publicly accessible, the user will be required to sign up to the application in order to use so that we can keep a track of who has access to the system and restrict this if needed in the future. Administration features of the application will be restricted however to account with the admin role within the database which can only be assigned by other admins. |
| TECHNICAL REQUIREMENTS |
| * The frontend client of the DD5e explorer will be built using angular JS. Angular JS is a JavaScript framework used across the industry for developing saleable and maintainable frontend client application which can be deployed stand-alone but can interact with external API’s using simple HTTP CRUD requests. * The backend API of DD5e explorer will be built in Express JS. Express JS is a Node backend framework for creating API routes with JavaScript. The benefit of Express JS is that it allows for creating and managing middleware to ensure our routes are access restricted where required. We also have access to a great unit testing library called Jest. Using Jest we can mock functions and procedures to ensure we only are testing one specific function within our codebase. * The frontend and backend applications will also be built using TypeScript instead of plain JavaScript because it is preferred in industry due to it being much easier to discovery bugs and issues during coding alongside its ability to ensure that the data being passed between the application is of the type we’d expect. Angular also automatically prevents XSS for in the event that some JavaScript code is read from the database. * The frontend’s UI will be created using a library called Angular Material. This is build off of Google’s material specification and is highly accessible and used across the web for it’s amazing properties. * The database which our API will be interacting with will be a mySQL database. The choice of an SQL database was that unlike Mongo for example, SQL is relational and as such we can relate user accounts to favourited items for example. The database will be interacted with using a secure SQL library within node package manager which will ensure our data is sanitised before being sent the api to prevent SQL injection. * The application will be hosted at <https://atwd2-ws269879.remote.ac> on the hosting platform <https://plesk.remote.ac/> provided by the client. I have been assured that the server supports NodeJS for my backend as well as all the newest security updates have been applied to the system. * Passwords will be stored as hashes in the database to ensure they are unreadable in the event of a database breach. Hashing algorithm used will be bcrypt and will also be salted.   *Considerations:*   * *I choice of technology stack for this project is effective as I work with this exact technology stack in my current employment and as such have a vast knowledge of the languages and frameworks with will be used within the project. I have built frontends in angular and backends in node/express in the past.* * *The use of Angular Material will ensure that this project is very accessible as well as developed quickly and efficiently as all the styling and accessibility has been developed and tested by a third party.* * *This project has a very simplistic aim of providing a user with the ability view statistics and details of d&d features and characters. As such this should be fairly simple while also providing a challenge as the extended features such as administration, authentication and expiry, and site statistics will add complexity.* * *Deploying an angular project should be very simple and AngularJS has a build command which will convert all files into a web accessibly format of HTML and JS files.* * *Deploying the API may be difficult as I have not experienced deploying to a node project to a plesk server before* |
| API INTEGRATION |
| I am going to be using a third party API for this project as previously stated. This API is called dnd5eapi and can be found at <http://www.dnd5eapi.co/> . The API will be directly from the client as there is no need to check authentication of the user as the API is not under my control. These calls will be done asynchronously as to prevent UI thread blocking and to ensure that the user can still use the application while details are loaded from the API.  The additional API I will create will require authentication which will be done using authentication cookies as well as JWT’s. This API needs to be authenticated as it deals with interaction to the database and we don’t want to allow anyone to delete and edit user records. The use of cookie based sessions are okay for a small project like this. |
| LEGAL, MORAL AND ETHNICAL CONCERNS |
| The app will store first and last names of user’s who have registered to the system alongside their emails and hashed passwords. This will mean that in the event of a database breach the list of emails and associated names would be exposed. However, this level of identifiable data is minimal.  All efforts will be made to ensure the project is accessible to as much of the populous as possible however due to personal skill and time constraints no attempt for alternative languages will be provided. Extra attention to screen readers and mobile devices will be given. |