# Database Rationale

The chosen database for the application is MongoDB Atlas. MongoDB Atlas is a cloud-hosted, document-oriented NoSQL database that aligns well with the technical requirements and structure of the D&D Toolkit application. Below is the technical rationale for the decision.

## Schema Flexibility

The document database allows for a less rigid structure. This is helpful for the application since the various tables will include data that frequently has missing values and nested structures. The nested structures are especially useful as they allow the storage of more complex data in an easy to use and read fashion. For example the ability scores can be stored as one attribute that has multiple key-value pairs for Strength, Dexterity, etc.

## Non-Relational Data

The data being used for the application is generally not related. While it can be useful to join data such as a user with their character sheet or include what user entered a specific log, this does not require intense interaction between tables in the database. For the MVP, the user will only be allowed to have one character at a time, so the character data will be directly stored with the user collection.

## Cost Effective

MongoDB Atlas is cloud-native and fully managed. There is no need to set up a server or handle the hosting details. This helps reduce development time and lower overhead costs. The platform also has a generous free tier which should be more than sufficient for the MVP. It is also scalable if more data requirement become necessary.

# Database Structure

The database for the application is made up of 4 collections. Two are used to persist data entered by users, while the other two are used to store data pertaining to random generation.

## Users

{

"\_id": ObjectId("64f2d1b8e4a13c28d7e4a820"),

"name": "Austin Glass",

"email": "austin@example.com",

"username": "austinglass01",

"password": "hashedpassword123",

"ability\_scores": {

"strength": 14,

"constitution": 13,

"dexterity": 12,

"intelligence": 16,

"wisdom": 10,

"charisma": 8

},

"skill\_proficiencies": ["Stealth", "Arcana", "Investigation"],

"level": 3,

"saving\_throws": ["Dexterity", "Intelligence"],

"armor": "Studded Leather",

"unarmored\_defense": false,

"shield": false,

"hit\_points": 21,

"character\_name": "Kael the Quiet",

"image\_path": "/images/characters/kael.png"

}

### Usage

The users data is used for two key purposes. First it allows authentication for the application. This will allow the application to carry the user context between pages using React Context. Since the user will be logged in, the application can retrieve their character data for the built in character sheet. The application will take the data that is stored with the user (ability scores, skill proficiencies, level, saving throws, armor, unarmored defense, shield, hit points, character name, and image path) and calculate the values to display using the D&D 5e official rules. From there, the application will display the results.

## Adventure Logs

{

"\_id": ObjectId("64f2d25be4a13c28d7e4a831"),

"type": “Session Recap”,

"title": "The Goblin Caves",

"description": "The party entered the dark goblin tunnels and encountered several traps before defeating the goblin shaman.",

"date": ISODate("2025-07-06T14:30:00Z")

}

### Usage

The adventure log is responsible for maintaining the key features within the campaign. This includes locations, NPCs, session recaps, and quests. The adventure log holds all of the data using the type attribute to determine which page it will be used to populate. Each of the adventure log pages will get the data from the database and then display a title, description, and entry date.

## Random NPCs

{

"\_id": ObjectId("64f2d2cbe4a13c28d7e4a845"),

"first\_name": "Thalindra",

"last\_name": “Moonshadow”

"image": "/images/npcs/thalindra.png"

}

### Usage

The Random NPCs collection holds the options available for the random NPC generator. This data is not added to or altered after the application is started. The application will select 3 random documents and pull one attribute from each document to generate a completely random and unique NPC.

## Random Monsters

{

"\_id": ObjectId("64f2d312e4a13c28d7e4a855"),

"name": "Shadow Drake",

"image": "/images/monsters/shadow\_drake.png",

"xp": 700,

"ref\_stat\_block": "https://open5e.com/monsters/shadow-drake"

}

### Usage

The Random Monsters collection holds the options available for the random encounter generator. This data is not added to or altered after the application is started. At the time an encounter is generated, the application will select random documents and apply the business rules to determine if the selection is valid.