Assignment 1 Note Taking App

Assignment 1.1

Introduction:

The purpose of assignment 1.1 is to plan and design the database and UI for a note-taking app. My note-taking app is going to be centred around gaming. The purpose of 1.1 phase is not to write code but rather to gather information, organize data and create a visual prototype of what the UI is going to look like before implementation of the code. This includes deciding on entities and their relationships, fields and the constraints of the database. I am also going to be sketching out wireframes of the dashboard, notes editor and login pages.

1: Database Schema and Diagram:

MongoDB is going to be the database I will be using for the assignment where it will have two collections one for the Users and one for the notes each user makes.

Users: Stores information about each user for login and note access

```
{
  id: ObjectID,
  name: String,
  email: String,
  password: String,
  createdAt: date,
  updatedAt: date
}
```

Notes: The content of the application where each note belongs to an individual user and includes text, tags and some attachments.

```
{
   Id: ObjectID,
   userID: ObjectId,
   title: String,
   content: String,
   tags: [String],
   isPinned: Boolean,
   isArchived: Boolean,
   priorty: String, // “urgent”, “done”, “normal” // colour coded to red, green and white
   attachments: [
       {
```

fileUrl: String,

fileType: String,

uploadedAt: date

}

],

createdAt: Date,

updatedAt: Date

}

Tags:

These are embedded in notes as an array of strings. Tags include the game titles, genres, platforms which will allow the user to filter and organize their notes.

Attachments:

These are embedded within the notes to simplify retrieval. These include screenshots, short game clips or reference files providing some each note context.

Relationships + ERD diagram:

- Users → Notes: One-to-many
- Notes → Attachments: One-to-many
- Notes → Tags: One-to-many

Some Justification for my design choices:

MongoDB / NoSQL:

- Notes taking apps from my experience have attachment and tag options so embedding them reduces joins and queries
- MongoDB has flexible schema so it allows for different note structure

Tags / Attachments:

- Faster retrieval of the notes when content is tightly coupled
- With gaming notes tags like game titles, genres or platform are small strings so no need for separate collection

Indexes:

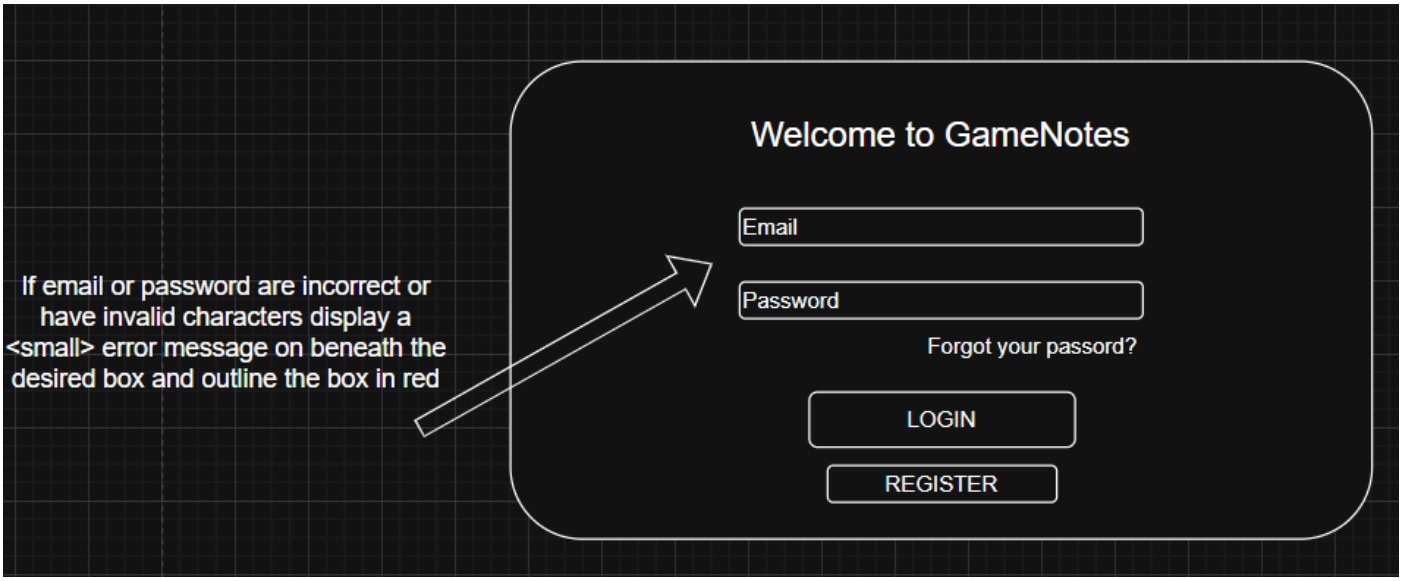
- Index userID on Notes for fast dashboard retrieval
- Index tags for searching and filtering by game or genre

Scalability:

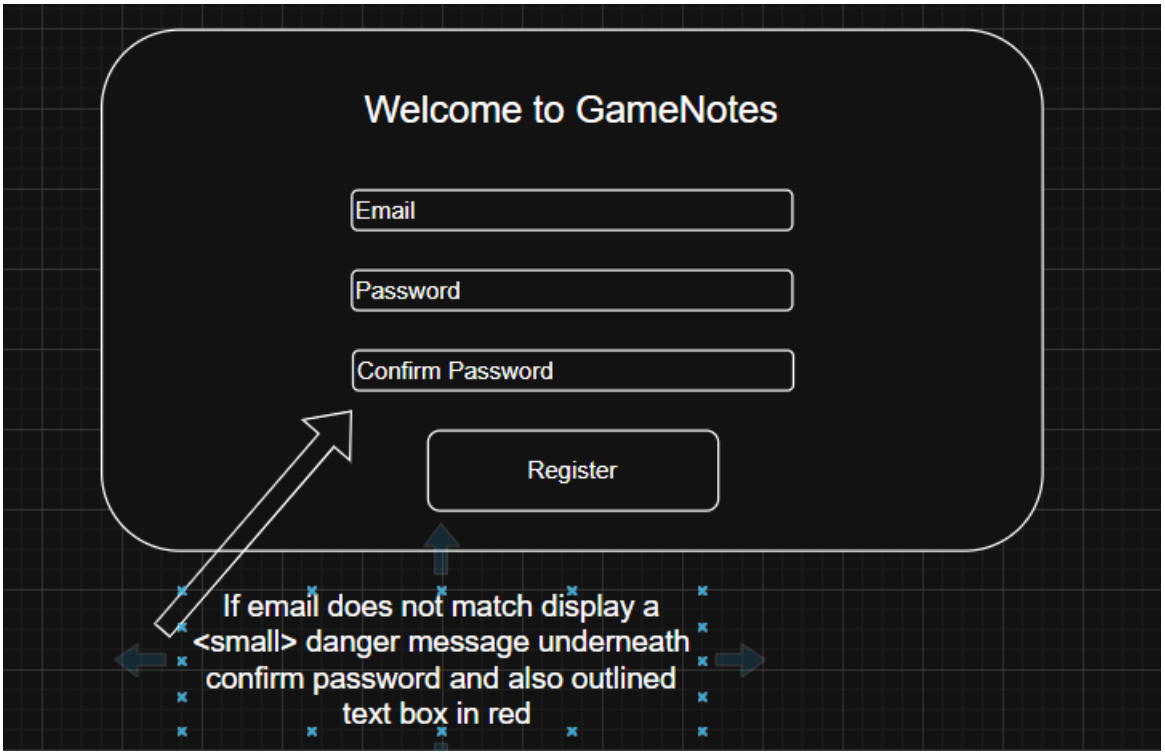
- Embedding works well for personal notes per user

2: UI Protoypes

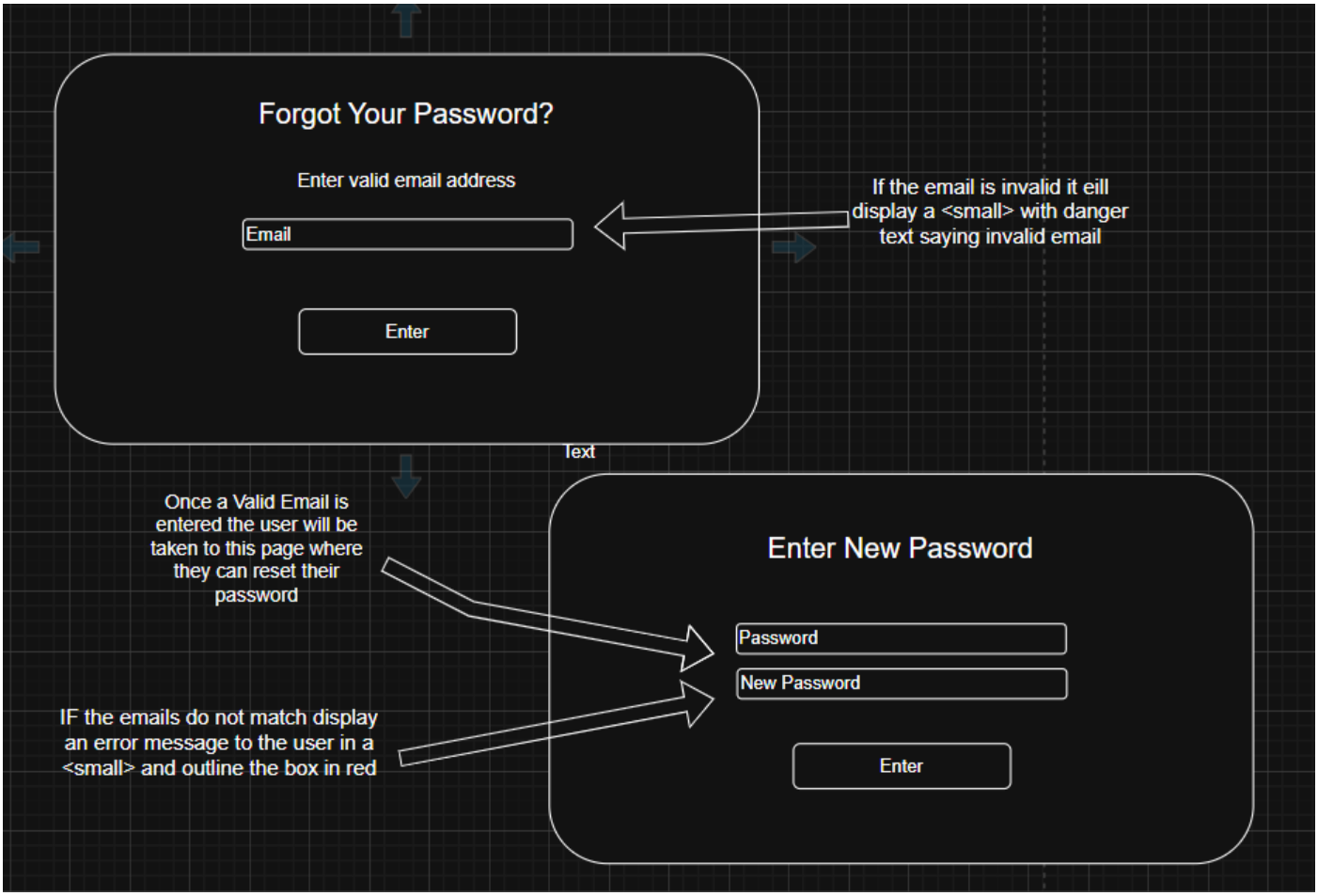
Login Page



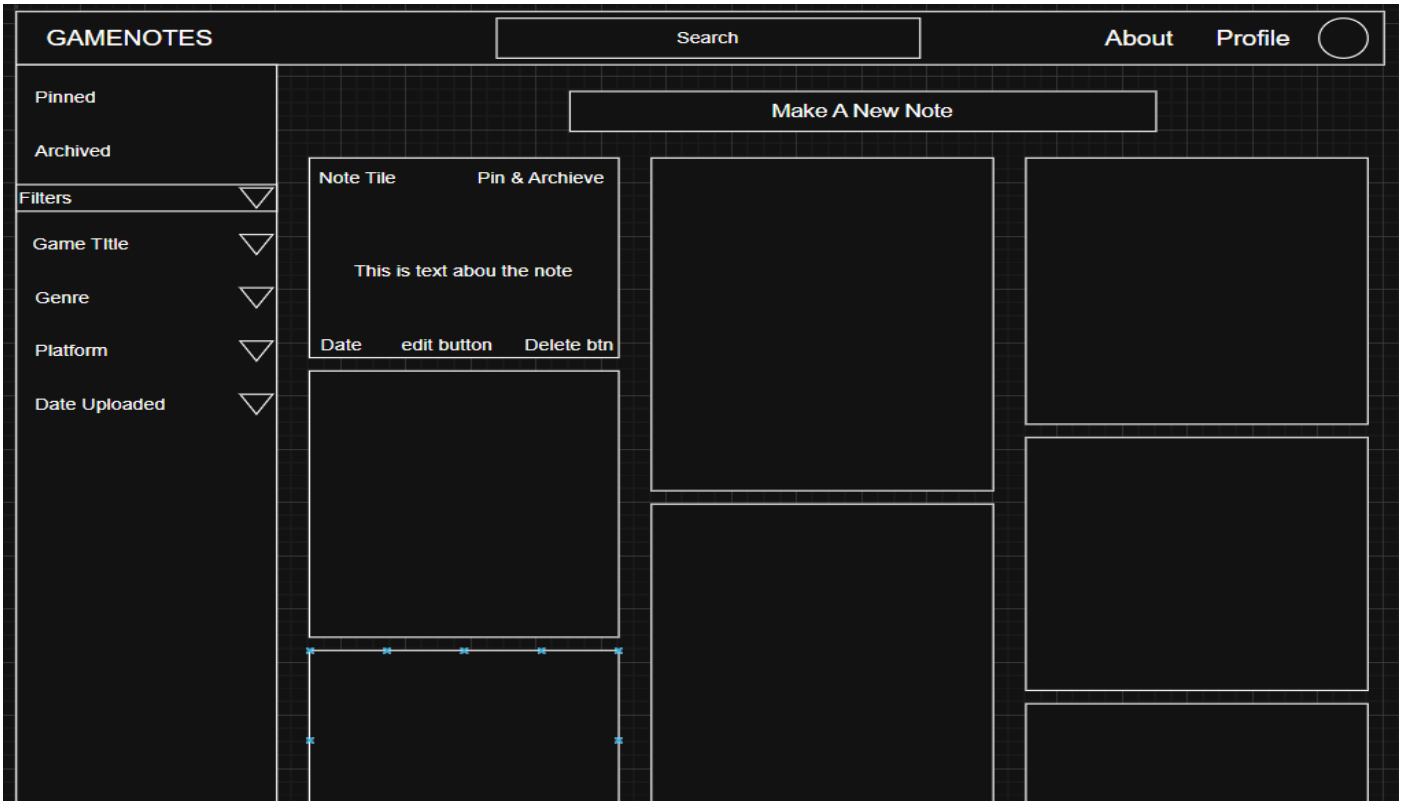
Register Page if register button is clicked



Forgot Password Page:



DashBoard



On the dashboard we have the top navigation bar that can be use to search notes, access the about and profile pages alongside a sidebar dedicated to filtering the notes and organising them. Users can filter through pinned and archive status, game title, genre, platform and date uploaded. The central area is the

notes display in a scrollable card based layout, where each card includes a title, pin and archive buttons, body of the note, date, edit button and delete button. I think this layout ensures efficient navigation.

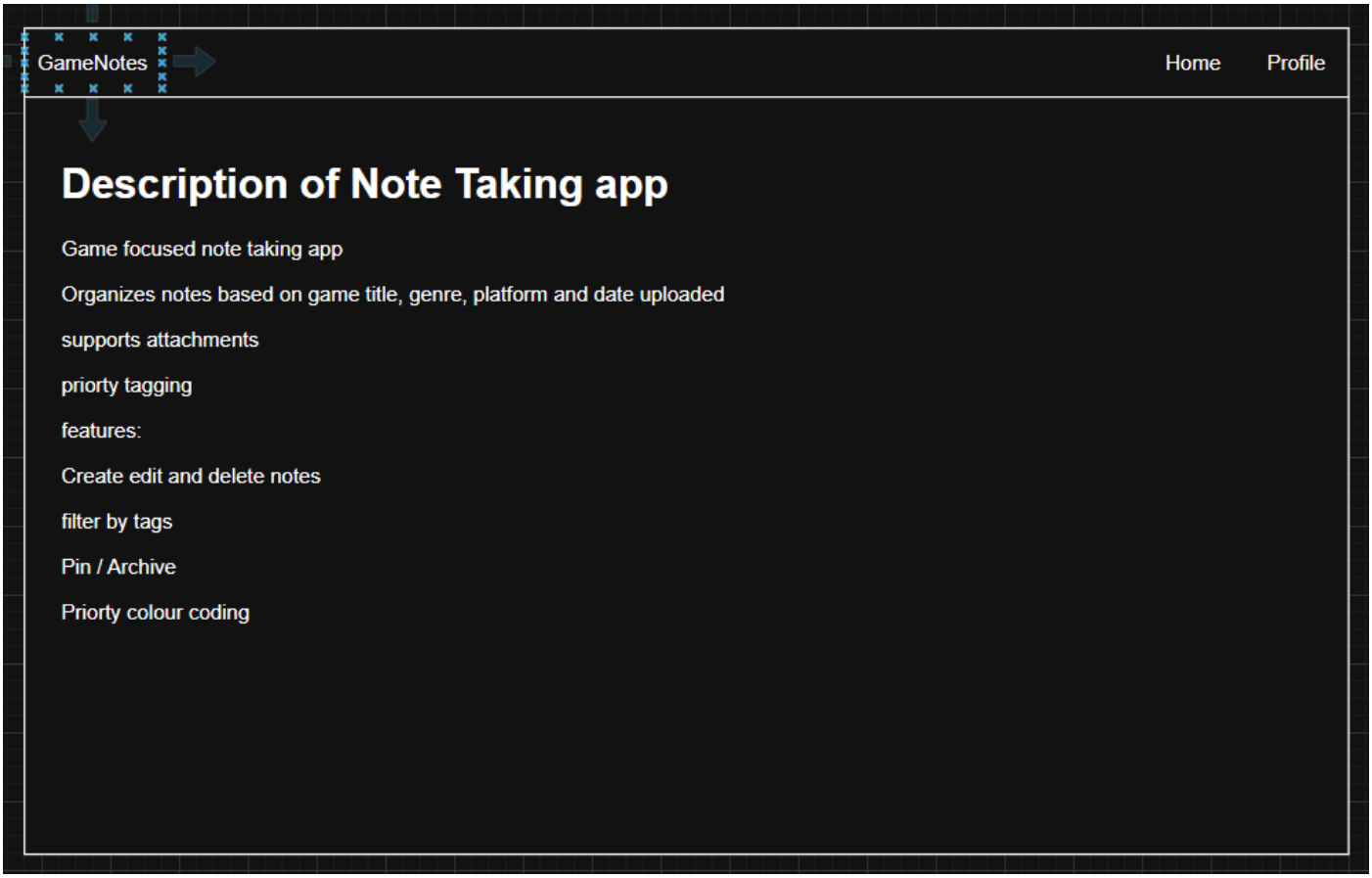
Edit Note page

When a user clicks the edit note button the note will enlarge itself and blur out the rest of the webpage. Within in this view the user is able to change the title, body text, tags, attachments, priorty status, Pin it or archive it. The user can choose to save their changes or cancel them which will both bring the user back to the dashboard.

New Note

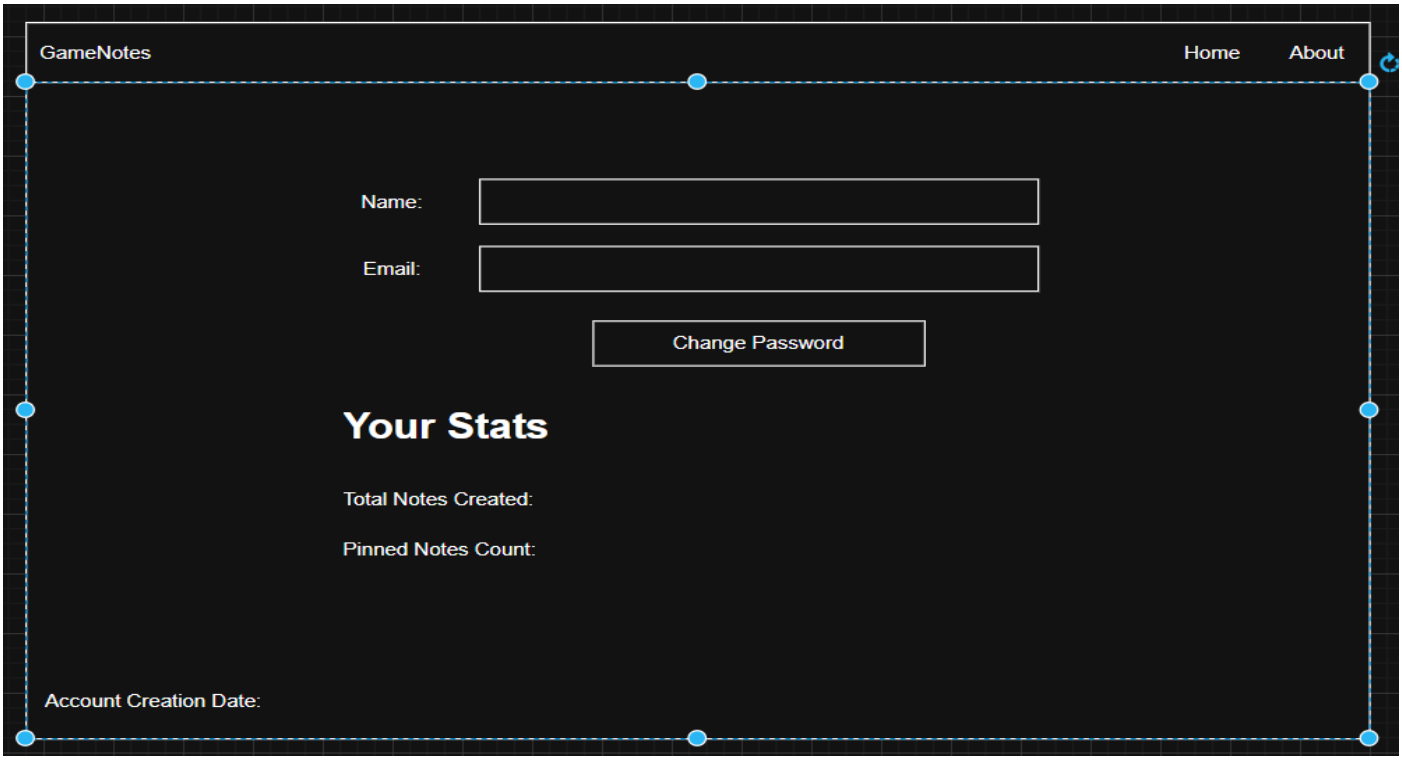
The new note page is very similar to the edit note page but instead you are creating a whole new instance of a note in the database using a post method instead of a put. The only thing that is different otherwise is the title.

About page



I thought it would be nice to have an about page just to make it feel more like an actual application. This page is just to the user what the app does and how it does it.

Profile Page



The profile page just has the users Name and email address on it and if they wish they can change their password. By clicking that button the user is brought to change password page, Also on this page there is some stats about the users Notes created and total notes made. Finally there is an Account creation date at the bottom.