Functional Requirements:

1. Search for characters by name, abilities, or teams.
2. View character profiles with images, bios, and timelines.
3. Filter characters by attributes (e.g., heroes, villains, anti-heroes).
4. Search comics by title, series, or release date.
5. Save comics to "favorites" or "to-read" lists.
6. Provide personalized comic recommendations.
7. User authentication using OAuth.
8. Dashboard summarizing user preferences and saved items.

Non-Functional Requirements:

1. **Performance:** Fast response times for search queries (<2 seconds).
2. **Scalability:** Handle a growing database of characters and comics.
3. **Usability:** Intuitive UI with responsive design.
4. **Security:** Secure user authentication and data storage.

Low-fidelity, Non-interactive, Mobile-First, Responsive GUI design/mockup (using figma):

Figma Link: https://www.figma.com/design/A85U9xDvjrntzoF4rn1IvB/Capstone?node-id=0-1&t=0bT4mYa8d1hwhFPY-1

Database Diagrams:

One database for comics and characters: {


```
"characters": [
  {
    "id": "1009368",
    "name": "Iron Man",
    "description": "A wealthy industrialist and genius inventor...",
    "image": "http://example.com/ironman.jpg",
    "affiliations": ["Avengers", "Stark Industries"]
  }
],
"comics": [
  {
```

```
      "id": "001",
      "title": "Avengers #1",
      "issueNumber": 1,
      "creators": ["Stan Lee", "Jack Kirby"],
      "releaseDate": "1963-09-01"
    }
  ]
}
```


And another database for users:


```
{
  "_id": "user123",
  "name": "Emily",
  "email": "emily@example.com",
  "savedCharacters": [
    { "id": "1009368", "name": "Iron Man" },
    { "id": "1009369", "name": "Thor" }
  ],
  "savedComics": [
    { "id": "001", "title": "Avengers #1", "issueNumber": 1 }
  ],
  "preferences": {
    "theme": "dark",
    "language": "en"
  },
  "collections": [
    {
      "id": "col001",
      "name": "Iron Man Collection",
      "comics": [
        { "id": "001", "title": "Avengers #1", "issueNumber": 1 }
      ]
```

```
    },
    {
      "id": "col002",
      "name": "Thor Collection",
      "comics": [
        { "id": "002", "title": "Thor #1", "issueNumber": 1 }
      ]
    }
  ]
}
```

Json web service input and output for each RESTful endpoint:

1. **GET /characters?name=Iron+Man**

   Input:
   json
   ```
   {
     "name": "Iron Man"
   }
   ```
   Output:
   json
   ```
   {
     "id": "1009368",
     "name": "Iron Man",
     "description": "A wealthy industrialist and genius inventor...",
     "image": "http://example.com/ironman.jpg"
   }
   ```

2. **POST /save-comic**

Input:

json

```
{
  "userId": "123",
  "comicId": "001"
}
```

Output:

json

```
{
  "message": "Comic saved successfully."
}
```

Architectural Stack:

Frontend: React, TailwindCSS.

Backend: Node.js

Database: MongoDB for user, comic, and character data

API Integration: Marvel API (free to use with an account)

Authentication: OAuth 2.0 for user authentication.

Hosting: Vercel (frontend), AWS (backend).