**Phase 1: *Read the project statement and find ideas on the rules and functionalities***

Description:

Online music store, to buy Albums, Singles or Merch. We can also see an Artist/Band page.

Products:

* Albums
* Singles
* Merch

Categories:

* Product type: each of the product, so we can filter them out. Merch will have subcategories:
  + Posters
  + Clothing
  + DVDs
  + Miscellaneous
* Genres:
  + Blues
  + Classical
  + Country
  + Electronic
  + Holiday
  + International
  + Jazz
  + Latin
  + Pop/Rock
  + Hip Hop/R&B
  + Reggae
  + Other
* Artists: each Artist/Band will have its own page with every product associated to it
* New releases: Highlight new addition to the store
* Featured Products: Showcase a random product per day.

Functionalities:

1. Product Listing:
   * Filter products based on the selected categories: Product Type (Album, Single, Merch -> type of merch), Genre, Artist
   * Implement pagination or infinite scrolling for a smooth browsing experience.
2. Product Details:
   * Provide detailed information for each product: title, artist (link to artist page), track listing, price, quantity, and an image.
   * For Merch: type (Posters, Clothing, DVDs, Miscellaneous), date.
3. Artist Pages:
   * Create dedicated pages for each artist or band, displaying information like the artist's biography, discography, and associated products.
4. New Releases Section:
   * Highlight newly added products to keep users informed about the latest additions to the store.
5. Featured Products Section:
   * Showcase a randomly selected product each day on the home page as a featured item.
6. Admin Panel:
   * Admin dashboard to get to each category for CRUD actions.

**Phase 2*: Analyze the needs by making an app mockup, a use case and a CDM/LDM if we choose a db-first approach or a class diagram if we choose a code-first approach. We also have to set up a git flow.***

1. Mockup with Figma
2. Use Case
3. CDM
4. LDM

**Phase 3*: We have to model and design the program DB.***

* Use LDM to create DB
* Add some data to test

**Phase 4*: The back-end part would be a REST API with a multi-layers architecture. It's supposed to be built in C#.***

* Type manuscript notes
* Set GitHub up
* Follow notes to build API
  + Connection string: "Server=localhost;User=Aura;Password=1234;Database=aurhapsody;port=3306"
  + Ef scaffold : dotnet ef dbcontext scaffold "Server=localhost;User=Aura;Password=1234;Database=aurhapsody;port=3306" Pomelo.EntityFrameworkCore.MySql -o Model --context-dir ../Data.Context/ -c AuRhapsodyDBContext -d -f

**Phase 5*: Develop the UI based on the mockup. We have to use REACT JS with Typescript.***

**Phase 6*: Test the program***