**E-commerce Rails Project Planning Phase**

**Description of the E-commerce Site**

**Business Name: Bookstore**

**Business Description:**

Bookstore is a small, independent bookstore with a mission to connect readers with books tailored to their tastes and interests. Established five years ago, Bookstore has grown steadily by emphasizing personalized customer service and offering a carefully curated selection of books across a variety of genres. The business employs a team of 15 passionate staff members who are knowledgeable about literature and dedicated to helping customers discover new reads. Currently, Bookstore sells its products primarily through an in-store experience, fostering a cozy and community-focused environment with shelves stocked with classic literature, contemporary fiction, and niche genres.

In response to increasing demand for online shopping, Bookstore is now looking to expand its reach with a robust e-commerce platform, allowing it to serve book lovers nationwide. This platform will enable Bookstore to connect with a broader audience while continuing to support its local community and customer base.

**Product Offering:**

Bookstore will feature a wide selection of books in its online store, including fiction, non-fiction, educational materials, children's books, self-help, and rare collectibles. The store also plans to offer unique book bundles, curated collections, and limited-edition prints for collectors and dedicated readers.

**Target Demographic:**

The intended customer base for Bookstore's online presence includes:

**Primary Audience:** Adults aged 18-50 who are passionate about reading, including college students, young professionals, parents, and general book enthusiasts.

**Secondary Audience:** Book collectors, bibliophiles, and academic professionals searching for rare or specialty books.

The platform is designed to appeal to readers who appreciate quality, variety, and personalized recommendations. The store will also target those who enjoy exploring books based on genre, author, and curated themes or collections.

**Key Features Planned for the E-commerce Site**

**Search and Filter Options:** Users can search for books by author, genre, publication date, and rating. Advanced filters will allow customers to tailor searches to their specific interests.

**User Reviews and Ratings:** Readers can leave reviews and rate the books they’ve purchased, creating a community-driven aspect for the store. These reviews will provide valuable insights for future customers, helping them make informed choices.

**Personalized Recommendations:** Based on past purchases or browsing history, Bookstore will offer tailored book recommendations.

**Community and Engagement:** Bookstore plans to host an online blog featuring articles, author interviews, and book recommendations, creating a lively space for book discussions and discovery.

# 

# **Database Structure Description**

**Tables**

**Author**

Purpose: Stores information about book authors for displaying author details on book pages.

**Columns:**

**name (string):** The author's name.

**bio (text):** A brief biography of the author.

**AdminUser**

Purpose: Manages admin-level users who can oversee site content and functionality.

Columns:

**email (string):** The admin user's email address.

**encrypted\_password (string):** Encrypted password for secure access.

**remember\_created\_at (datetime):** Timestamp for the "remember me" functionality.

**reset\_password\_sent\_at (datetime):** Timestamp for password reset requests.

**reset\_password\_token (string):** Token for resetting the password.

**ActiveAdmin::Comment**

Purpose: Tracks comments or notes made by admins within the backend interface.

Columns:

**author\_type (string):** Type of author making the comment.

**body (text):** Content of the comment.

**namespace (string):** Namespace for categorizing comments.

**resource\_id (integer):** ID of the resource commented on.

**resource\_type (string):** Type of resource being commented on.

**AuthorShip**

Purpose: Joins authors to books in a many-to-many relationship.

Columns:

**author\_id (integer):** Foreign key to the Author table.

**book\_id (integer):** Foreign key to the Book table.

**Book**

Purpose: Stores information about each book in the store.

Columns:

**date (date):** Date-related field, possibly for cataloging or release.

**description (text):** Description or synopsis of the book.

**publication\_date (date):** When the book was published.

**title (string):** Title of the book.

Review

Purpose: Contains user-submitted reviews for books.

Columns:

**content (text):** Content of the review.

**rating (integer):** Numerical rating given to the book.

BookGenre

Purpose: Connects books to genres in a many-to-many relationship.

Columns:

**content (text):** Information or notes about the genre association (if needed).

**rating (integer):** Possible rating or ranking related to the book-genre pairing.

Genre

Purpose: Defines book genres for filtering and categorization.

Columns:

**name (string)**: The name of the genre.

**ActiveStorage::VariantRecord**

Purpose: Stores data about specific file variants (e.g., resized images).

Columns:

**variation\_digest (string):** Digest for tracking the variant's unique attributes.

**ActiveStorage::Blob**

Purpose: Stores metadata for files uploaded to the site.

Columns:

**byte\_size (integer):** Size of the file in bytes.

**checksum (string):** Checksum for verifying file integrity.

**content\_type (string):** File’s content type (e.g., image/jpeg).

**filename (string)**: Original filename.

**key (string):** Key for locating the file in storage.

**metadata (text):** Additional metadata.

**service\_name (string):** The name of the storage service used.

**ActiveStorage::Attachment**

Purpose: Links blobs to their associated records in the app.

Columns:

**name (string):** Name of the attachment.

**record\_type (string):** Type of record the attachment is associated with.

**Associations**

**Author and Book**

Association: Many-to-Many

Through Table: AuthorShip

Description: A book can have multiple authors, and an author can have written multiple books. The AuthorShip table acts as a join table to manage this relationship with author\_id and book\_id.

**Book and Genre**

Association: Many-to-Many

Through Table: BookGenre

Description: A book can belong to multiple genres, and each genre can encompass multiple books. BookGenre serves as the join table to manage this association with book\_id and genre\_id.

**Book and Review**

Association: One-to-Many

Description: A book can have multiple reviews associated with it, but each review is linked to one specific book. This allows users to submit individual reviews per book, with book\_id acting as the foreign key in the Review table.

**AdminUser and ActiveAdmin::Comment**

Association: One-to-Many

Description: Each admin user can write multiple comments (through ActiveAdmin) on various resources, helping manage content or providing notes in the admin interface. Each comment stores a reference to the AdminUser who created it.

**ActiveAdmin::Comment and Resources (Polymorphic)**

Association: Polymorphic One-to-Many

Description: ActiveAdmin::Comment can be associated with different resource types, such as books, genres, or authors. The polymorphic fields (resource\_id and resource\_type) allow a single ActiveAdmin::Comment to be linked to various models within the application.

**ActiveStorage::Attachment and Associated Models (Polymorphic)**

Association: Polymorphic One-to-One or One-to-Many

Description: ActiveStorage::Attachment allows various records, such as books or authors, to have attached files (e.g., images or documents). The fields name, record\_type, and record\_id manage this relationship, allowing each model to have one or more attachments as needed.

**ActiveStorage::Blob and ActiveStorage::Attachment**

Association: One-to-One

Description: ActiveStorage::Blob stores file metadata (such as content type and byte size) for each file attachment. Each attachment points to a specific blob, enabling the application to retrieve file information or modifications efficiently.

**ActiveStorage::VariantRecord and ActiveStorage::Blob**

Association: One-to-One

Description: Each ActiveStorage::Blob can have associated file variants (such as resized images), stored in ActiveStorage::VariantRecord. The variation\_digest links each variant to a specific version of the original file, enabling the storage and retrieval of multiple versions per blob.