ABSTRACT

This project is kind of prototype of recipe website. Users are able to upload recipe (with their requirements such as image, name etc.) and leave a comment under specific recipes.

Also there's an admin panel which provide the manager accept or decline user comments or suggested recipes.

Contents

Introduction

This website is created to help people find some recipes for either making their new year's eve party warmer or delicious dinner.

Also the main concept of engender this site is make a site which has more user-friendly UI than the others, ruin the ordinariness and provide ease of usage.

Used Technologies

1. ASP.NET Framework:
   ASP.NET Web Forms for building the user interface.

2. C# Programming Language:
   Primary language for server-side logic and backend development.

3. Visual Studio IDE:
   Used for the development, debugging, and deployment of the ASP.NET application.

4. HTML5 and CSS3:
   Used for structuring and styling the web pages.

5. JavaScript:
   Enhanced client-side interactivity and user experience.

6. Database Management System:
   Microsoft SQL Server.

Requirements

Data Requirements

Data Schema

➔ Contact Table:
  ContactID
  ContactPerson
  ContactTitle
  ContactMail
  ContactContent

➔ Recipes Table:
  RecipeId (PK)
  RecipeName
  RecipeGredients
  RecipeInstruction
  RecipeImage
  RecipeOwnerMail
  RecipeOwner
  RecipeCategoryId
  RecipeState
  Viewed

➔ Food Rate Table
  FoodId (PK)
  RateAverage
  RateQuantity

➔ Category Table:
  CategoryId (PK)
  CategoryName
  CategoryQuantity
  CategoryImage

➔ Food Table:
  FoodId (PK)
  FoodName
  Ingredient
  Recipe
  FoodImage
  FoodDate
  CategoryId

➔ Comment Table:
  CommentId (PK)
  CommentName
  CommentMail
  CommentDate
  CommentRate
  CommentApproval
  CommentContent
  FoodId
  Viewed

Data Entites
"Recipe" – "Category" – "Food" – "Comment"

Data Constraints

	Recipe	Category	Food	Comment
Primary Key Constraint	RecipeId	CategoryId	FoodId	CommentId
Foreign Key Constraint	-	-	CategoryId	FoodId
Check Constraint	Viewed	-	-	Viewed

Functional Requirements

Recipe Management
User is able to add new recipes to the website & database

Category Management
User is able to select category of unique recipes
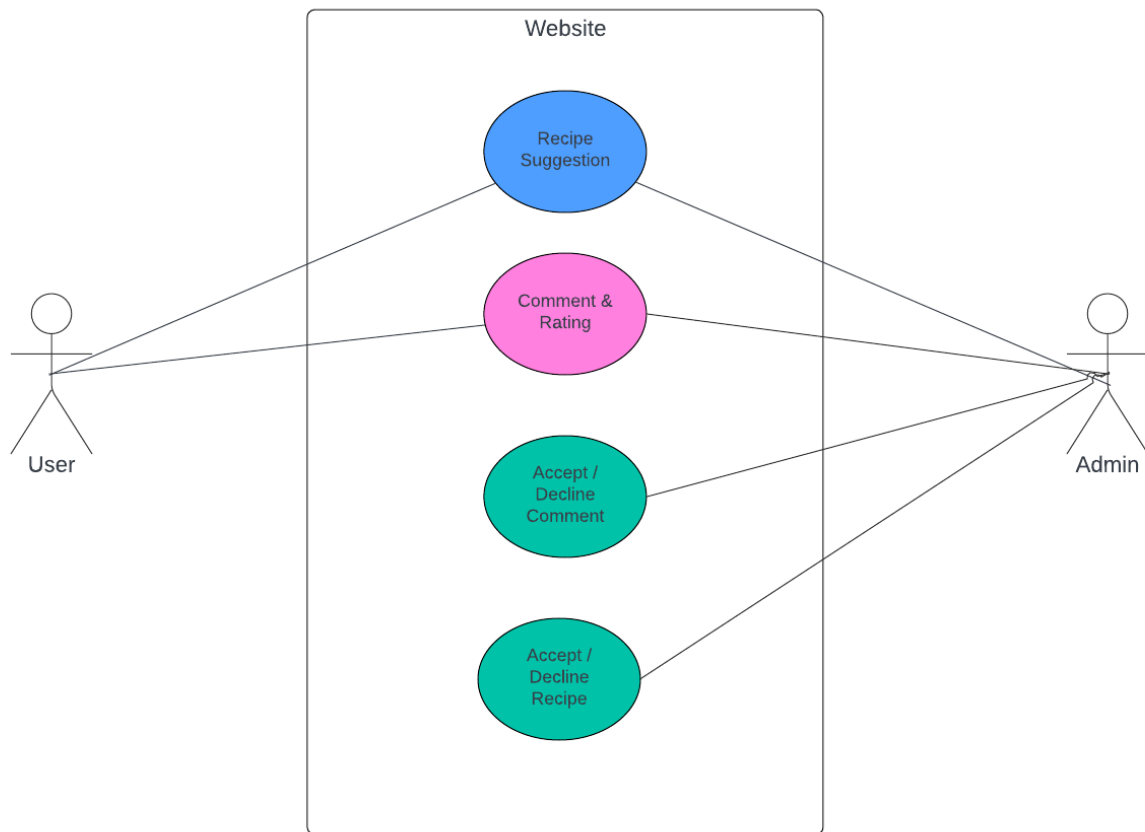
Rating and Reviews
User is able to leave comment and star rating (e.g from 1 to 5)

Admin Functionality
Admin is able to accept or decline left comments or suggested recipes

Application Modeling Steps

UML Use Case Diagram

UML Class Diagram

Categories

+ CategoryId: int
+ CategoryName: string
+ CategoryQuantity: int
+ CategoryImage: string

Recipes

+ RecipeId: int
+ RecipeName: string
+ RecipeGredients: string
+ RecipeInstruction: string
+ RecipeImage: string
+ RecipeOwnerMail: string
+ RecipeOwner: string
+ RecipeCategoryId: int
+ RecipeState: boolean
+ Viewed: boolean

+ CopyRecipeToFoods(): trigger
- transmitInfo(): void

+1

+1..*

Foods

+ FoodId: int
+ FoodName: string
+ Ingredient: string
+ Recipe: string
+ FoodImage: string
+ FoodDate: datetime
+ CategoryId: int

+ FoodTableTrigger(): trigger
+ CreateFoodRateEntity(): trigger

+1..*

Comments

+ CommentId: int
+ CommentName: string
+ CommentMail: string
+ CommentDate: datetime
+ CommentRate: int
+ CommentApproval: boolean
+ CommentContent: string
+ FoodId: int
+ Viewed: boolean

+ UpdateFoodRate(): trigger

+1

+1

Food Rate

+ FoodId: int
+ RateAverage: float
+ RateQuantity: int