|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Epic | User Story | Acceptance criteria |
| 1. | Admin login to system so that he can perform admin actions on restaurants | 1.Admin can access the login form and provide Credentials.  2.Admin accesses dashboard so that he can manage restaurant | 1.1 Admin should be able to access the login form.  1.2 Admin should be able to provide email and password  1.3 On successive 3 login failures account should be locked  1.4 On failed login, show the error message  1.5 Admin should be able to login with google account  2.1 Admin can be able to view list of restaurants  2.2 Provision to assess the details of restaurant  2.3 Link should be able to delete the restaurant  2.4 If any open orders are there for the restaurant, admin cannot delete the restaurant |
| 2. | Admin login to system so that he can perform admin actions on users | 1.Admin can access dashboard so that he can manage users  2. Button/Menu item should be available to add new restaurant so that he can add new restaurant.  3.Admin adds new users to the system so that new user | Admin should be able to view users list.  Link should be provided to deactivate the user.  2.1 Admin should be able to navigate to Add-user form.  3.1 Admin should be able to enter user details.  3.2 System should check for duplicate emails. |
| 3. | The restaurant owner login to the app so that he can manage the Restaurant. | 1. Restaurant owners can access the dashboard to their restaurants list. | * 1. Restaurant owners should be able to view Restaurants.   2. Owner should be able to add new Restaurants. |
| 4. | Restaurant Owner can manage menu items. | 1. Restaurant owners can access the dashboard to their restaurant’s menu list. 2. Restaurant Owners should be able to manage the menu list. | * 1. Owners should be able to view Menu items of a restaurant.   2. System should validate mandatory fields.   3. owners should be able to navigate to add menu list.   4. Initial values should be set zero   5. The owner should be able to delete menu items only if no active orders are placed.   6. The owner should be able to add Cuisine Photos if required.   7. He should be able to remove duplicate items. |
| 5. | The User should be able to login to the app so that he can Order food. | 1.User can access the login form and provide Credentials.  2.User can access Restaurants available on the app according to the location. | 1.1 User should be able to access the login form.  1.2 On successive 3 login failures account should be locked  1.3 On failed login, show the error message.  2.1 User can see the restaurants list.  2.2 System should show a list of restaurants near the entered locations  2.3 User should be able to sort /search by distance, rating, cuisine type. |
| 6. | Users should be able to filter menu items based on food preferences. | 1. Users can filter the menu based on preferences like cuisine, veg, non-veg, price range. | 1.1 User should be able to apply multiple filters.  1.2 Filtered results should reflect immediately without page reload.  1.3 System should retain applied filters till session is active. |
| 7. | Users should be able to place orders for menu items. | 1.User can select menu items and place an order for a given restaurant.  2. User can select the pay method and place the order. | 1.1 User should be able to add items to cart.  1.2 User should be able to view-cart, edit cart items, and proceed to checkout.  2.1 User should be able to select delivery address and make payments.  2.2 System should generate an order confirmation with order id.  2.3 On successful order, the restaurant should be notified. |
| 8. | Update order status upon delivery. | 1. Upon successful delivery the order status must be updated. | 1.1   Delivery agent or system should be able to update the status of the order to “Delivered”.   1. Users should be notified via app notification and email once the order is delivered 2. Order history should reflect the updated status. |