**Quest 1:** Design an E-commerce system. Which includes the following set of features.

• Users can login/Sign Up using their contact no.

• User can place order

• Cart System

• Product listing

• Product review, rating and wishlist

**Code:**

|  |
| --- |
|  |
| public class UnitOfWork : IDisposable  {  private DatabaseContext context; {  private GenericRepository<User> customerRepository;  private GenericRepository<User> userRepository;  private GenericRepository<Product> productRepository;  private GenericRepository<Order> orderRepository;  private GenericRepository<OrderProduct> orderProductRepository;  }  public GenericRepository<User> CustomertRepository  public GenericRepository<User> UserRepository  {  get  {    if (this.customerRepository == null)  if (this.userRepository == null)  {  this.customerRepository = new GenericRepository<User>(context);  this.userRepository = new GenericRepository<User>(context);  }  return customerRepository;  return userRepository;  }  } |
| **Quest 2:** Design a polling system like Telegram. Where users have multiple options to vote from. Allow users to vote and show the live vote count.  **Example:**  **Question:** What should be the timing for tomorrow’s class.  Options:   1. After 5PM - 80% 2. Before 5PM- 20%   **Code:**  var dataPoints = [  { label: "After 5PM", y: 0 },  { label: "Before 5PM", y: 0 },  ]  var chartContainer = document.querySelector('#chartContainer');  if (chartContainer) {  var chart = new CanvasJS.Chart("chartContainer", {  animationEnabled: true,  theme: "theme2",  data: [  {  type: "column",  dataPoints: dataPoints  }  ]  });    chart.render();  }    Pusher.logToConsole = true;    // Configure Pusher instance  const pusher = new Pusher('PUSHER\_APP\_KEY', {  cluster: 'PUSHER\_APP\_CLUSTER',  encrypted: true  });    // Subscribe to poll trigger  var channel = pusher.subscribe('poll');    // Listen to vote event  channel.bind('vote', function(data) {  dataPoints = dataPoints.map(dataPoint => {  if(dataPoint.label == data[4].name[0]) {  dataPoint.y += 10;  }    return dataPoint  });    // Re-render chart  chart.render()  }); |  |  |