Bài tập: WCF Chat.

1. Tạo project Windows Store: Chọn New Project -> Templates -> Visual C# -> WCF -> WCF Service Application
2. Cài đặt EntityFramework nếu như chưa có.
3. Tạo 2 Entity Model như sau:
   1. class Chat
      1. Id: int, PrimaryKey, Identity(1,1)
      2. Content: string
      3. Username: string, ForeignKey nối sang bảng User
      4. SentTime: DateTime
   2. class User
      1. Username: string, PrimaryKey
      2. Password: string
4. Tạo DataContext với DbSet để kết nối dữ liệu
5. Tạo DataInitializer với 1 vài dữ liệu mẫu để demo và test.
6. Trên service interface IService, tạo OperationContract sau:
   1. bool Login(string Username, string Password)
   2. string Register(string Username, string Password)
   3. List<Chat> GetAllChat()
   4. String SendChat(string Content, string Username)
7. Trên WCF service Service.svc, implement lại interface vừa tạo và xử lý các phương thức:
   1. bool Login(string Username, string Password): Kiểm tra nếu tồn tại User có trùng Username và Password trong cơ sở dữ liệu thì trả về true, còn lại trả về false.
   2. string Register(string Username, string Password): Thêm mới vào cơ sở dữ liệu User có Username và Password như trên parameter. Xử lý lỗi trùng Username và các lỗi nhập liệu khác và trả về thông báo tương ứng.
   3. List<Chat> GetAllChat(): Trả về danh sách tất cả các Chat có trong CSDL
   4. String SendChat(string Content, string Username): Thêm mới 1 chat vào CSDL, SentTime được set theo thời gian của máy chủ.
8. Tạo MVC Project, consume WCF, sử dụng WCF thay cho kết nối database để thực hiện các chức năng sau:
   1. Trên Home, tạo Action và View Login, có thể sử dụng lại Template Create với entity User từ WCF. Khi người dùng nhập đúng thông tin đăng nhập và Login, gọi đến WCF service (thông qua hàm gọi trên Action Login có phương thức HttpPost) kiểm tra thông tin đăng nhập, nếu đúng thì tạo Session[“LoginUser”] = Username, sau đó cho chuyển sang Action / View Chat, nếu sai thì trả về thông báo lỗi lên giao diện Login.
   2. Trên Home, tạo Action và View Register, sử dụng Template Create với entity User từ WCF. Khi người dùng Submit thông tin, dùng WCF Service đăng ký thêm mới người dùng và trả kết quả về màn hình. Nếu đăng ký thành công thì tạo Session[“LoginUser”] = Username và đẩy sang Action / View Chat.
   3. Trên Home, tạo Action và View AllChat, hiển thị các chat dưới dạng bảng 2 cột: Username và Content. Mỗi khi chạy action, thông qua WCF lấy tất cả Chat về hiển thị lên bảng trên View.
   4. Trên Home, tạo Action và View Chat, trên View cho 1 iframe trỏ đến View AllChat để hiển thị tất cả nội dung chat hiện có. Bên dưới là 1 textbox Message và nút Send để nhập và gửi nội dung Chat mới lên WCF. Khi chưa đăng nhập (sử dụng Session[“LoginUser”] cho phần đăng nhập, khi chưa đăng nhập thì session này bằng null hoặc empty), đẩy người dùng sang action Login trong HomeController. Khi đã đăng nhập, hiện View và khi người dùng bấm nút Send, gửi thông điệp ở ô Message cùng với Session[“LoginUser”] lên WCF để nhập thêm mới nội dung Chat vào cơ sở dữ liệu.
   5. Dùng Javascript, cứ mỗi 3 giây lại load lại iframe AllChat trên view Chat một lần để hiển thị toàn bộ nội dung chat.
   6. Thay form thường thành form Ajax cho đoạn nút và textbox gửi Chat mới.
9. Kết luận: WCF có lợi thế về tạo kết nối dữ liệu nhanh giữa client và service, hỗ trợ dễ dàng kết nối và truyền tải message giữa 2 phần. Tuy nhiên, đây không phải là cách tối ưu để làm ứng dụng Chat do truyền tải dữ liệu trên đường truyền nhiều, gặp giới hạn thông tin truyền và chậm dần khi lượng thông tin chat tăng lên. (Với phạm vi nội dung bài tập này) Do đó có thể dùng phương pháp khác hoặc tạo 1 ứng dụng Chat đơn giản hơn với WebAPI ở buổi sau.