

# Advanced Programming

Dr. Shirin Baghoolizadeh



**Telegram:**  
**@AmirArsalanYavari**

**Email:**  
**amirarsalan.yavari@ec.iut.ac.ir**

## مقدمه

در این پروژه شما ملزم به استفاده از زبان **C++** و تفکر شی‌گرائی + **QT framework** هستید؛ همچنین بایستی موارد ذکر شده در پروژه را رعایت کرده و یک برنامه‌ی پیام‌رسان طراحی نمایید.

## شرح پروژه:

### 1. ورود یا ثبت‌نام

برای ورود به سیستم کاربر یا باید صفحه‌ی **login** را مشاهده کند یا به صفحه‌ی **register** منتقل شود. در صفحه‌ی **login** کاربر با استفاده از نام‌کاربری و رمز عبور وارد برنامه شود؛ در صفحه‌ی **register** نیز با استفاده از نام‌کاربری، شماره تلفن، رمز عبور، تاریخ تولد، نشانی ایمیل و... یک حساب کاربری ایجاد نماید (دقت فرمایید که نشانی ایمیل و شماره تلفن باید متمایز باشد).

### 2. ایجاد صفحه‌ی ارسال پیام

برنامه‌ی شما باید قابلیت این را داشته باشد که هر فرد بتواند افراد را بر اساس نام یا آیدی و یا حتی از روی شماره‌ی تلفن شخص یافته و به او پیام دهد. لازم به ذکر است اگر کاربری به کاربر دیگری پیام داد صفحه‌ی پیام باید از آن پس برای هر دو نمایش داده شود و از آن پس الزاماً قرار نیست برای پیام دادم مراحل ایجاد صفحه‌ی چت را پیاده‌سازی کنند.

### 3. ایجاد گروه

در این قسمت قرار است کاربر تعدادی از کاربران را انتخاب کند و همچنین یک اسم برای گروه انتخاب کند. از این پس هر پیامی که داخل آن صفحه ارسال شود برای تمامی کاربران قابل مشاهده است و تمامی کاربران امکان ارسال پیام در آن صفحه را دارند.

### 4. ایجاد کانال

همانند گروه است با این تفاوت که فقط سازنده‌ی کانال و یا افرادی که اون مشخص می‌کند حق ارسال پیام دارند.

### 5. تنظیمات کاربری

در این قسمت منوی پروفایل هر کاربر به او نشان داده می‌شود و کاربر باید بتواند داده‌های مرتبط با خود را تغییر دهد؛ اطلاعاتی نظیر نام‌کاربری، شماره‌ی تلفن، سال تولد (الزامی نیست)، عکس پروفایل و درکل هر پارامتر دیگری که در دیگر نرم‌افزارهای اجتماعی دیده می‌شوند...

همچنین سطح دسترسی به اطلاعات را برای دیگر کاربران مشخص کند؛ برای مثال مشخص کند کدامیک از گروه کاربران اجازه‌ی مشاهده‌ی عکس پروفایل یا شماره‌ی تلفن او را دارند.

لازم به ذکر است هر کاربر باید بتواند اطلاعات قابل مشاهده از دیگر کاربران را ببیند.

## 6. گراف سطح سه

در صفحه‌ی پروفایل هر کاربر گرافی تا سطح سه از ارتباطات آن کاربر نشان‌دهید؛ تمام نودهای این گراف که امکان اتصال به هم را دارند متصل نشان دهید. دقت فرمایید که گراف باید واضح باشد.

## 7. موارد دارای نمره‌ی اضافه

استفاده از الگوریتم‌های رمزگذاری RSA و AES256 در ذخیره و ارسال داده‌ها (50 نمره)  
در صورتی که بتوانید با API ایمیل کار کنید، 50 نمره امتیازی خواهید گرفت. (برای مثال در هنگام تراکنش‌ها اطلاع رسانی صورت گیرد یا در هنگام ورود به سیستم هر کاربر با یک کلید رندوم authenticate شود)

اگر از Socket استفاده کنید و برنامه‌ی شما بتواند تحت وب هم کار کند به میزان 100 نمره‌ی اضافی، نمره‌ی بیشتری دریافت خواهید کرد.

اگر پروژه‌ی خود را داکرایز کردید که بتواند روی هر سیستمی که داکر دارد اجرا شود 30 نمره‌ی اضافی خواهید گرفت.

موارد فوق در صورتی شامل نمره خواهند بود که پروژه‌ی شما تقریباً **90 درصد** کامل باشد. (سعی نکنین بخش‌هایی از پروژه رو نزنین که با این قسمت‌های امتیازی نمرش رو جبران کنید!)

## 8. نکات پایانی

در طراحی کلاس‌ها به **encapsulation** و **data hiding** بسیار توجه کنید و حتماً حتماً آنها را رعایت کنید، همچنین در مواردی که ارث‌بری انجام می‌دهید به مواردی ظیر **abstraction** نیز دقت و در صورت نیاز انجام دهید.

تمیزی و reusable بودن کد شما بسیار مهم است و از نمره‌ی اضافی ویژه‌ای برخوردار است؛ بدین منظور می‌توانید به این [لینک](#) مراجعه نمایید.

برنامه‌ی شما تحت هیچ‌شرایطی نباید مموری‌لیک داشته باشد. پس مدیریت منابع را واجب‌تر از نون شب بدوینید :

برنامه‌ی شما باید مولتی‌ترد (multy thread) باشد و درست کار کند. بدین صورت که قبل از نمایش صفحه‌ی ورود یک server app دارید که به ازای هر اینستنس جدید باید یک اینستنس جدید ایجاد شود. اینترفیس لازم برای کار کردن با server app برعهده‌ی خود شماست.

در انجام این پروژه ملزم هستید از git استفاده کنید و موارد زیر را حتما رعایت کرده باشید چرا که در غیر اینصورت میزانی از نمره‌ی پروژه‌ی شما کسر خواهد شد.

1. کامیت‌ها باید روند اجرا و تکمیل پیوسته‌ی پروژه را نشان دهد.
2. با github باید کار کنید.
3. بعد از رفع دیپندنسی‌های اولیه اعضای هر گروه ملزم به ایجاد برنچ و درنهایت مرج کردن آن در زمان مناسب با برنچ اصلی می‌باشند.
4. کامیت‌ها باید بامعنی باشند؛ بدین منظور می‌توانید به این [لینک](#) مراجعه فرمائید.
5. برای پروژه‌ی خود در github باید یک readmi مناسب (کوتاه و مختصر و در عین حال کامل) بنویسید.
6. افراد گروه پس از اطلاع رسانی گروهشان به یک ریپازیتوری اضافه می‌شوند و بایستی تمام پروژه را در آن repo ادامه دهند. (مشارکت همه‌ی اعضای گروه در پل پروژه‌ی شما الزامی است)

• تحویل این پروژه در سه فاز انجام می‌شود:

o فاز اول

- طراحی کلاس ها و توابع مورد نیاز با توجه به اصول شی گرای و الگو های طراحی.
- راه اندازی گیت.

o فاز دوم

- پیاده سازی توابع و کلاس های طراحی شده.

o فاز سوم

- پیاده سازی رابط گرافیکی و بخش های مربوط به شبکه.

زمان تحویل پروژه ۵ام تیرماه خواهد بود؛ زمان ارائه ی پروژه توسط تمام اعضای گروه متعاقبا اعلام می گردد.

### **توجه توجه!!!**

**دقت** فرمائید این یک مدل ساده شده از یک پیام رسان است و این دست از برنامه ها باید با استفاده از سوکت و با الگوی کلاینت سروری اجرا شوند.