



## فازهای پروژه

پروژه شامل ۳ فاز اصلی به شرح زیر است. جدول زیر عناوین هر فاز و وظایف مربوط به آن را خلاصه می‌کند:

جدول ۱: فازهای پروژه و وظایف هر فاز
عنوان فاز و توضیحات
فاز ۱: دریافت اطلاعات دریافت تعداد دانشجویان، گرفتن نمره هر دانشجو از ورودی و ذخیره آن‌ها (مثلاً در یک list)
فاز ۲: محاسبات اولیه محاسبه شاخص‌های اولیه کلاس مانند میانگین (average) نمرات و تعیین بالاترین و پایین‌ترین نمره
فاز ۳: ارزیابی و خروجی محاسبه تعداد دانشجویان قبول شده (نمره‌های ۱۰ و بالاتر) و مردود شده، و نمایش خلاصه نتایج محاسبات به صورت خروجی خوانا

سلام به همه دوستان عزیز! امیدوارم تا اینجا ترم از یادگیری پایتون لذت برده باشید. حالا وقتشه که آموخته‌ها تون رو در یک پروژه پایانی سرگرم‌کننده و کاربردی به کار بگیرید. به عنوان دستیار آموزشی این درس، می‌خوام به پروژه پایانی باحال بهتون معرفی کنم که هم مباحث مقدماتی پایتون رو پوشش می‌ده، هم شما رو با استفاده عملی از کتابخانه‌های NumPy و Pandas درگیر می‌کنه و در نهایت به رابط گرافیکی ساده با Tkinter یا PyQt براتون می‌سازه. نتیجه نهایی چیزی شبیه به برنامه دسکتاپ کوچیک خواهد بود که می‌تونه به فایل داده (مثلاً یک CSV از نمرات دانشجوها) رو بخونه، تحلیل‌های ساده روش انجام بده و خروجی رو به شکل دوستانه داخل به پنجره گرافیکی نشون بده. این یعنی قراره هم با داده کار کنیم، هم با واسط کاربری - بدون نیاز به خروجی گرفتن در ترمینال! سناریوی پروژه: تصور کنید می‌خوایم یک برنامه بنویسیم که به یک استاد یا دستیار آموزشی کمک کنه نمرات پایانی دانشجوها رو سریع‌تر محاسبه کنه. داده‌های خام نمرات به صورت یک فایل CSV (یا Excel ساده) در اختیار مونه که توش مثلاً نمرات تکالیف، کوئیزها و امتحان پایان‌ترم هر دانشجو موجوده. برنامه قراره این فایل رو بخونه، نمره نهایی هر دانشجو رو حساب کنه، معادل حرفی (A, B, C, ...) براش تعیین کنه، و همین‌طور به خلاصه آماری از کل کلاس بده. تازه، قراره به رابط گرافیکی هم بسازیم که با زدن چند تا دکمه، فایل نمرات رو وارد کنه و نتایج رو نمایش بده. ایده کار الهام گرفته از ابزارهای واقعی‌ای هست که همین کار رو انجام می‌دن. به نظرتون جالب نیست که خودتون همچین برنامه‌ای بسازید؟ این پروژه به فازهای مشخصی تقسیم شده تا کارتتون قدم به قدم جلو بره. طی دو هفته با انجام این فازها، در نهایت یک برنامه کامل خواهید داشت که می‌تونید بهش افتخار کنید. هر فاز اهداف خودش رو داره و قابل انجام در بازه زمانی کوتاه است، پس نگران حجم کار نباشید.

### فاز ۱: خواندن داده و محاسبات ابتدایی (مبانی پایتون)

در فاز اول، می‌خوایم هسته منطقی برنامه رو بدون واسط گرافیکی پیاده کنیم. تمرکز این بخش روی مباحث پایه‌ای پایتونه: متغیرها، انواع داده‌ای ساده، حلقه‌ها، شرط‌ها، توابع و کار با فایل. اهداف این فاز عبارت‌اند از:

- خواندن فایل داده نمرات: یک فایل CSV به شما داده می‌شه که شامل ستون‌های StudentName (نام دانشجو)، AssignmentScore (نمره تمرین‌ها)، QuizScore (نمره کوئیزها) و FinalExam (نمره امتحان پایانی) هست. ابتدا باید برنامه‌ای بنویسید که این فایل رو بخونه. می‌تونید از قابلیت‌های پایه‌ای پایتون مثل تابع open و پیمایش خط به خط استفاده کنید یا حتی از کتابخانه داخلی csv بهره ببرید. هدف اینه که ساختار کلی خوندن فایل رو یاد بگیرید و داده‌ها رو داخل یک ساختار مناسبی مثل لیست یا دیکشنری ذخیره کنید.
- محاسبه نمره نهایی هر دانشجو: فرض می‌کنیم نمره نهایی هر دانشجو بر اساس وزنی از این سه بخش محاسبه می‌شه. برای مثال، ۳۰٪ از نمره مربوط به تمرین‌ها، ۲۰٪ مربوط به کوئیزها و ۵۰٪ مربوط به امتحان پایانی باشه (شما می‌تونید این ضرایب رو در برنامه به صورت ثابت تعریف کنید). بر این اساس، یک تابع پایتونی بنویسید که با گرفتن نمرات هر بخش برای یک دانشجو، نمره نهایی تجمعی اون دانشجو رو محاسبه کنه. (مثلاً اگر دانشجویی ۸۰ از تمرین‌ها، ۷۰ از کوئیزها و ۹۰ از امتحان گرفته باشه، نمره نهایی‌ش  $\approx 90 \times 0.5 + 70 \times 0.2 + 80 \times 0.3$  خواهد شد).
- تبدیل نمره عددی به حرفی (Grade Conversion): حالا وقتشه یک شرط‌گذاری ساده پیاده کنیم. برای جذاب‌تر شدن کار، می‌تونیم نمره نهایی ۱۰۰ نمره‌ای هر دانشجو رو به حرف تبدیل کنیم. یک مقیاس ساده می‌تونه این باشه: اگر نمره نهایی  $\leq 90$  باشه "A"، اگر بین ۸۰ تا ۸۹ باشه "B"، بین ۷۰ تا ۷۹ "C"، بین ۶۰ تا ۶۹ "D" و پایین‌تر از ۶۰ "F" در نظر بگیریم. پس تابع یا قطعه کدی بنویسید که یک نمره عددی (مثلاً ۸۵) رو ورودی بگیره و براساس این محدوده‌ها حرف متناظرش رو خروجی بده. (دقت کنید این قسمت صرفاً چند if-elif ساده‌ست و پیچیدگی خاصی نداره).
- محاسبات آماری کلی (اختیاری در این فاز): اگه تا اینجا کار رو سریع انجام دادید، می‌تونید چند تا خروجی کلی هم از کل کلاس بگیرید. مثلاً میانگین نمره نهایی کلاس، بالاترین و پایین‌ترین نمره، یا چند تا آمار ساده دیگه. این بخش اختیاریه اما به درک بهترتون از داده کمک می‌کنه.

در پایان فاز ۱، باید یک اسکریپت پایتون داشته باشید که بتونه فایل نمرات رو بخونه و نتایج اولیه رو مثلاً در ترمینال چاپ کنه. مثلاً خروجی می‌تونه چیزی شبیه این باشه: (فقط نمونه)

```
1 Ali: Final Score = 75 -> Grade = C
2 Sara: Final Score = 92 -> Grade = A
3 ...
4 Class Average = 78.5
5 Max = 92, Min = 60
```



## پروژه پایانی مبانی برنامه سازی پایتون

مهم نیست خروجی رو دقیقاً چه شکلی نشون بدید؛ فعلاً خوانا بودنش کفایت می‌کنه. حتماً برنامه‌تون رو روی فایل نمونه‌ای که داریم تست کنید تا مطمئن بشید همه چیز درست کار می‌کنه.

**نکات راهنما برای فاز ۱:** اگر تا حالا با فایل CSV کار نکردید، نگران نباشید. فایل CSV در واقع همون فایل متنی که مقادیرش با کاما جدا شدن. می‌تونید هر خط رو بخونید و با متد `split(',')` به بخش‌های جداگانه تقسیم کنید. همچنین استفاده از کتابخونه `csv` پایتون کارتون رو راحت‌تر می‌کنه. حتماً برای تبدیل رشته به عدد (مثلاً نمرات که به صورت متن خونده میشن) از `int()` یا `float()` استفاده کنید. در مورد درصد وزنی نمرات هم می‌تونید ضرایب `0.3`، `0.2` و `0.5` رو در متغیرهای جداگانه بذارید که اگر خواستیم تغییر بدیم، راحت باشه.

### فاز ۲: استفاده از کتابخانه‌های NumPy و Pandas

حالا که منطق پایه‌ای رو پیاده‌سازی کردید، وقتشه با ابزارهای قدرتمندتری آشنا بشید که کار با داده‌ها رو خیلی راحت‌تر می‌کنن. در این فاز تمرکز روی به‌کارگیری `Pandas` و `NumPy` هست تا عملیات فاز ۱ رو ساده‌تر و سریع‌تر انجام بدیم و داده‌ها رو بهتر مدیریت کنیم. اهداف فاز ۲ به طور مشخص:

- خواندن داده با `Pandas`: کتابخانه `Pandas` به شما این امکان رو می‌ده که فقط با یک خط کد، فایل CSV رو بخونید و داخل یک `DataFrame` (جدول داده) بریزید. مثلاً با `pd.read_csv("grades.csv")` می‌تونید تمام داده‌های نمرات رو یکجا بارگذاری کنید. این `DataFrame` مثل یه جدول اکسل یا بانک اطلاعاتی عمل می‌کنه و کلی تابع کمکی داره. حتماً این کار رو امتحان کنید و مثلاً با `print(df.head())` پنج سطر اول داده رو ببینید که مطمئن بشید فایل درست خونده شده.
- محاسبه نمرات با ابزارهای برداری: به جای اینکه برای هر دانشجو به صورت جداگانه حلقه بزنید، می‌تونید از قابلیت برداری (وکتورایز) در کتابخانه‌های `Pandas/NumPy` استفاده کنید. به عنوان مثال، می‌شه ستون‌های نمرات رو با ضرایبشون ضرب کرد و با هم جمع کرد تا ستون جدیدی به اسم `"FinalScore"` بدست بیاد. اگر `DataFrame` شما `df` باشه، می‌تونید چیزی شبیه این انجام بدید:

```
df["FinalScore"] = 0.3 * df["AssignmentScore"] + 0.2 * df["QuizScore"] + 0.5 * df["FinalExam"]
```

این خط به طور خودکار برای همه ردیف‌ها حساب می‌کنه و یک ستون جدید اضافه می‌کنه. دیدید چقدر کوتاه‌تر از کد فاز ۱ شد؟! در پس‌زمینه، `Pandas` داره از `NumPy` کمک می‌گیره تا این محاسبات برداری رو انجام بده.

- تبدیل به حرف با `NumPy/Pandas`: می‌تونید همون منطق `if`‌های فاز ۱ رو اینجا هم اعمال کنید، منتها راه بهتر استفاده از قابلیت `apply` در `Pandas` یا `mapping even`. مثلاً می‌تونید یک تابع پایتونی تعریف کنید که ورودی‌ش یک نمره نهایی باشه و خروجی‌ش حرف مربوطه. بعد با

```
df["FinalGrade"] = df["FinalScore"].apply(score_to_grade)
```

روی کل ستون `FinalScore` تبدیل انجام بشه. یا روش پیشرفته‌تر، تعریف یک دیکشنری از بازه‌ها و استفاده از `pd.cut` یا `np.select` برای دسته‌بندی نمرات به حروف. هر روشی را که دوست دارید امتحان کنید، ولی خروجی باید یک ستون جدید به اسم `"FinalGrade"` باشه که برای هر دانشجو حروف `C/B/A` ... داره.

- محاسبات آماری با `Pandas`: `Pandas` توابع خیلی کاربردی مثل `mean()`، `max()`، `min()` و حتی `describe()` رو داره که با اون‌ها می‌تونید به راحتی میانگین کل کلاس، بالاترین و پایین‌ترین نمره و سایر مشخصات آماری رو به دست بیارید. مثلاً `df["FinalScore"].mean()` یا `class_avg = df["FinalScore"].mean()` میانگین نمرات نهایی کلاس رو محاسبه می‌کنه. یا `df.describe()` یک جدول خلاصه از تعداد داده‌ها، میانگین، انحراف معیار و ... برای هر ستون عددی نشون می‌ده که خیلی به درد می‌خوره. حتماً یکی دو تا از این امکانات رو استفاده کنید و خروجی‌ش رو ببینید. حتی می‌تونید آمار توزیع حروف رو هم دریابید؛ مثلاً ببینید چند تا `A`، چند تا `B` و ... داریم (با `value_counts()` روی ستون `FinalGrade`).

- ذخیره خروجی نهایی (اختیاری): یک کار مفید اینه که نتایج نهایی رو دوباره توی یک فایل ذخیره کنیم. فرض کنید استاد می‌خواد یک فایل Excel داشته باشه که توش نمره نهایی و حرفی هر دانشجو آمده باشه. با `Pandas` می‌شه خیلی راحت این کار رو کرد:

خروجی CSV بگیرید. این قسمت رو انجام بدید تا مطمئن بشید خروجی‌های برنامه‌تون قابل نگهداری هست. (ضمن اینکه ما برای تصحیح خودکار شاید از همین فایل خروجی شما استفاده کنیم تا نمره‌دهی کنیم!)

در پایان فاز ۲، برنامه شما (هنوز در حالت غیربصری/ترمینال) باید بتونه: فایل نمرات رو بخونه، ستون‌های `FinalScore` و `FinalGrade` رو به داده اضافه کنه، چند خلاصه آماری مثل میانگین و ... حساب کنه، و یک فایل خروجی جدید حاوی نتایج نهایی ذخیره کنه. اگر این کارها رو از طریق توابع `NumPy/Pandas` انجام بدید، حتماً متوجه شدید که کدتون به شکل محسوسی کوتاه‌تر، تمیزتر و سریع‌تر شده. این دقیقاً قدرت استفاده از کتابخونه‌هاست!

**نکات راهنما برای فاز ۲:** قبل از کدنویسی، حتماً مستندات یا راهنمای سریع `Pandas` رو به نگاه بندازید تا با سینتکسش آشنا بشید. مثلاً بدونید چطور می‌شه به ستون‌های `DataFrame` دسترسی داشت یا یک ستون جدید اضافه کرد. اگر جایی به مشکل خوردید (مثلاً در نصب `NumPy/Pandas` یا خواندن فایل)، از دوستانتون یا ما کمک بگیرید. موقع استفاده از `to_excel`، شاید نیاز باشه کتابخونه `openpyxl` رو نصب کنید؛ اگه اروری دیدید، با `pip install openpyxl` برطرف می‌شه. در ضمن، نگران نباشید که چرا هم `Pandas` داریم هم `NumPy` - معمولاً `Pandas` کافیه، ولی خوبه بدونید که `Pandas` روی `NumPy`

```
df.to_excel("FinalGrades.xlsx", index=False)
```



## پروژه پایانی مبانی برنامه سازی پایتون

یا اگر Excel در دسرتون نصب داره، میتونید

```
3 df.to_csv("FinalGrades.csv", index=False)
4
```

ساخته شده و گاهی ترکیب این دوتا قدرت زیادی بهتر می‌ده. برای تمرین هم که شده، مثلاً می‌تونید لیست نمرات نهایی رو به numpy array تبدیل کنید و با np.mean یا np.max هم میانگین و ماکزیمم رو حساب کنید تا دستتون راه بیفته.

### فاز ۳: طراحی رابط گرافیکی (GUI) ساده

رسیدیم به بخش هیجان‌انگیز پروژه، یعنی اضافه کردن رابط کاربری گرافیکی! تا اینجا برنامه ما از طریق ترمینال کار می‌کرد؛ کاربر (مثلاً همان استاد یا شما) باید کد رو اجرا می‌کرد و خروجی‌ها رو در متن می‌دید. اما خیلی جذاب‌تر می‌شه اگر یه پنجره گرافیکی داشته باشیم با دکمه‌هایی مثل "بارگذاری فایل نمرات" و "محاسبه نتایج" و یک بخش که نتایج رو داخل خودش نشون بده. توی این فاز قراره با استفاده از کتابخانه Tkinter (یا اگر علاقه داشتید PyQt) یک واسط کاربری برای برنامه‌تون بسازید که تعامل باهاش راحت‌تر و قشنگ‌تر باشه. اهداف فاز ۳:

- ایجاد پنجره اصلی برنامه: با Tkinter می‌تونید خیلی سریع یک پنجره بسازید. مثلاً:

```
1 import tkinter as tk
2 root = tk.Tk()
3 root.title("Grade Analyzer")
4 root.geometry("600x400") # -> window size
```

این چند خط یک پنجره ۶۰۰x۴۰۰ با عنوان "Grade Analyzer" باز می‌کنه. می‌تونید عنوان و اندازه رو هر جور دوست دارید تنظیم کنید.

- دکمه برای انتخاب فایل: می‌خوایم وقتی کاربر روی یه دکمه کلیک کرد، یه پنجره انتخاب فایل باز بشه و فایل CSV رو ازش بگیره. خوشبختانه Tkinter این قابلیت رو داره. می‌تونید از tkinter.filedialog.askopenfilename استفاده کنید تا مسیر فایل رو دریافت کنید. مثلاً:

```
1 from tkinter import filedialog
2 file_path = filedialog.askopenfilename(title="Select file", filetypes=[("CSV files", "*.csv")])
```

این کد یک دیالوگ باز می‌کنه که فقط فایل‌های CSV رو نشون می‌ده و مسیر فایل انتخاب شده رو برمی‌گردونه. حالا می‌تونید این مسیر رو بدید به تابعی که تو فاز ۲ نوشتید (مثلاً به pd.read\_csv) و داده‌ها رو لود کنید. پس یک دکمه مثلاً با عنوان "باز کردن فایل نمرات" بسازید (tk.Button) و command اون رو متصل کنید به تابعی که این کار رو انجام می‌ده.

- دکمه یا رویداد محاسبه: وقتی فایل انتخاب و خوانده شد، مرحله بعدی محاسبه نتایج (نمره نهایی و ...) و نمایش اون‌هاست. شما می‌تونید بلافاصله بعد از لود کردن فایل محاسبات رو انجام بدید و نمایش بدید، یا یک دکمه جدا برای "محاسبه نتایج" در نظر بگیرید. برای سادگی، می‌شه همون لحظه که فایل انتخاب شد، محاسبات رو انجام داد. ولی برای انعطاف بیشتر، بد نیست یک دکمه "محاسبه" بذاریم که مثلاً اگر خواستیم قبلش تغییری در داده بدیم (بعداً شاید امکانات بیشتری اضافه کنیم) کار مفید باشه. خلاصه، تصمیم با خودتونه اما حتماً باید جایی در کدتون تابعی صدا بشه که کارهای فاز ۲ (محاسبه نمرات و آمار) رو انجام بده.

- نمایش نتایج داخل GUI: حالا مهم‌ترین قسمت رابط: خروجی نتایج رو داخل خود پنجره نشون بدیم. چند راه دارید: ساده‌ترینش اینه که از یک Text ویجت (tk.Text) یا حتی یک Label بزرگ استفاده کنید و متن نتایج رو اونجا قرار بدید. مثلاً می‌تونید رشته‌ای آماده کنید که شامل لیست دانشجویها و نمرات و حرف‌هاشون باشه (شبیه خروجی فاز ۱) و بعد با text\_widget.insert اون رو داخل تکست‌باکس بنویسید. یا می‌شه خلاصه‌آمار رو هم اضافه کنید. اگر دوست دارید یه مقدار حرفه‌ای‌تر بشه، می‌تونید از ttk.Treeview برای ساختن جدول استفاده کنید که داده‌ها رو مثل اکسل جدول‌بندی نشون بده. این کمی پیچیده‌تره ولی خروجی تر و تمیزتری می‌ده (اختیاریه کاملاً). حتی می‌تونید با کتابخانه‌های جانبی توی Tkinter چارت هم رسم کنید، ولی این رو هم می‌ذاریم جزو بخش اختیاری/پیشرفته.

- تجمع همه چیز: در نهایت باید دقت کنید که برنامه‌تون از لحظه‌ای که GUI رو اجرا می‌کنه (یعنی root.mainloop() در Tkinter)، همه چیز رو تحت کنترل رابط کاربری دربیاره. یعنی مثلاً وقتی دکمه "باز کردن فایل" کلیک شد، تابع مربوطه باید داده رو لود کنه و احتمالاً یک برچسب (Label) توی پنجره بذارید که اسم فایل انتخاب شده رو نشون بده که کاربر مطمئن بشه. بعد از اون یا با زدن دکمه "محاسبه" یا خودبه‌خود، نتایج در textBox یا جدول نمایش داده بشن. سعی کنید چیدمان (Layout) دکمه‌ها و جعبه‌متن‌تون مرتب و قابل فهم باشه. از pack یا grid برای نظم دادن استفاده کنید. مثلاً دکمه‌ها بالا باشن، خروجی پایین. یا هر ترتیب منطقی دیگه. هرچه رابطتون کاربرپسندتر باشه، بهتره – انگار دارید برای یه آدم غیرفنی این ابزار رو می‌سازید که فقط بتونه کلیک کنه و نتیجه رو ببینه.

در پایان فاز ۳، خروجی پروژه شما یک فایل py. اجرایی خواهد بود که وقتی ران بشه، یه پنجره گرافیکی باز می‌کنه. کاربر با استفاده از اون پنجره می‌تونه فایل grades.csv رو انتخاب کنه و نتیجه پردازش رو همونجا ببینه. مثلاً تصور کنید استاد ما برنامه رو باز می‌کنه، فایل grades.csv رو انتخاب می‌کنه، یه دکمه می‌زنه، و لیست دانشجویها با نمره و grade کنارشون ظاهر می‌شه؛ شاید پایین پنجره هم بنویسه "Class Average: 78.5, Max:92, Min:60". این دقیقاً چیزیه که می‌خوایم بهش برسیم. اگر برنامه‌تون رو اجرا کردید و همچین چیزی دیدید، تبریک می‌گم، شما یه اپلیکیشن پایتونی با رابط کاربری ساختید!



## پروژه پایانی مبانی برنامه سازی پایتون

نکات راهنما برای فاز ۳: طراحی GUI ممکنه در ابتدا کمی گیج کننده باشه، پس قدم به قدم پیش برید. اول مطمئن بشید می تونید به پنجره با به دکمه ساده بسازید. بعد همون دکمه رو طوری کنید که به پیغام یا مقداری متن رو داخل به Text widget نشون بده (این تست کمک می کنه مفهوم callback رو بفهمید). وقتی از این مطمئن شدید، برید سراغ وصل کردنش به کدهای Phase ۲. توصیه می کنم توابع اصلی خوندن داده و محاسبه رو که در Phase ۲ نوشته بودید، در یک فایل جدا یا بخش جدا نگه دارید (مثلاً توابع load\_data, compute\_final\_scores و ...). بعد در کد GUI فقط اون ها رو صدا کنید. اینجوری هم ساختار کدتون تمیزتر می شه هم تصحیح خودکار راحت تر می تونه منطق برنامه شما رو تست کنه. اگر با Tkinter احساس راحتی نمی کنید و قبلاً با PyQt کار کردید، استفاده از PyQt هم آذاده؛ ولی یادتون باشه PyQt کمی حجیم تره و تنظیمات اولیه بیشتری می خواد. Tkinter برای کار ما کفایت می کنه. در صورت استفاده از PyQt حتماً به توضیح بدید که چطور باید کدتون رو اجرا کنیم (مثلاً نیاز به نصب PyQt5 داره).

### جمع بندی فازها و ایده های خلاقانه (اختیاری)

تا اینجا سه فاز اصلی رو تعریف کردیم که انتظار می ره در مدت حدود دو هفته انجام شدن باشن. اگر از برنامه تون زودتر نتیجه گرفتید یا به نظرتون خیلی آسون بود، می تونید با افزودن چاشنی خلاقیت پروژه رو جذاب تر کنید. این موارد اختیاری هستن ولی انجامشون می تونه براتون تجربه خوبی باشه (و چه بسا امتیاز اضافه بگیرید):

- رسم نمودار: توزیع نمرات حرفی یا حتی خود نمرات عددی رو با یک نمودار ساده نشون بدید. مثلاً یک نمودار میله ای (bar chart) که تعداد A, B, C ... رو نمایش بده. برای این کار می تونید از کتابخانه Matplotlib استفاده کنید و نمودار رو داخل Tkinter Canvas با رسم کنید. اگر سخت به نظر میاد، حتی می تونید نمودار رو توی یک پنجره جدا باز کنید. داشتن نمودار به قشنگی پروژه اضافه می کنه.
- امکان ویرایش یا فیلتر داده: مثلاً کاربر بتونه انتخاب کنه که فقط دانشجویهای یک گروه خاص رو ببینه (البته داده های ما ممکنه گروه بندی نداشته باشه، ولی فرض کنید مثلاً امکان انتخاب "فقط دانشجویهای با Grade = A" برای نمایش وجود داشته باشه). این کار نیاز به تعاملات بیشتر در GUI داره (مثل ComboBox یا Entry برای فیلتر)، که اگر علاقه دارید می تونید اضافه کنید.
- شخصی سازی رابط کاربری: می تونید کمی رنگ و لعاب بدید. مثلاً رنگ پس زمینه پنجره رو عوض کنید، عنوان ها رو بولد کنید یا فونت رو بزرگ تر کنید که خوانا تر باشه. حتی می تونید به آیکون برای برنامه تنظیم کنید که موقع اجرا کنار عنوان پنجره نشون داده بشه. این چیزها شاید جزو الزامات فنی نباشه ولی حس خوبی به کاربر می ده که با به برنامه حرفه ای تر کار می کنه.
- پشتیبانی از فرمت های مختلف فایل: اگه حس کردید CSV خوندن رو مسلط شدید، می تونید قابلیت خوندن Excel (xlsx) رو هم اضافه کنید (pandas با pd.read\_excel این کار رو می کنه). یا حتی خروجی رو همزمان هم به Excel بده هم به CSV. این ها چیزهایی هستن که در دنیای واقعی برنامه ها دیده می شه.

خلاصه هر ایده جالبی به ذهنتون می رسه که فکر می کنید برنامه رو بهتر می کنه و در حد دانش این ترم قابل پیاده سازی، از امتحانش نترسید. پروژه شما مال شماست و خلاقیت شما حتماً مورد استقبال قرار می گیره.

### نحوه تحویل پروژه

لطفاً پروژه نهایی تون رو در قالب یک فولدر فشرده (zip) تحویل بدید. داخل این فولدر باید موارد زیر موجود باشه:

- کدهای پروژه: حداقل یک فایل پایتون که اجرای اصلی برنامه از اون شروع می شه (مثلاً main.py یا هر نام مناسبی). اگر کدهاتون رو بین چند فایل تقسیم کردید (مثلاً یک فایل برای GUI و یک فایل برای منطق)، مطمئن بشید همه اون ها رو ضمیمه کردید و برنامه به درستی اجرا می شه.
- فایل داده نمونه: همون فایل grades.csv (یا هر نامی که استفاده کردید) که برنامه باهاش کار می کنه رو هم قرار بدید. توجه: ممکنه ما داده های متفاوتی رو برای تصحیح تست کنیم، پس برنامه تون رو طوری بنویسید که وابسته به محتوای خاصی نباشه (مثلاً تعداد ردیف ثابت یا نام مشخصی). اما داشتن یک نمونه برای اینکه بدونیم فرمت ورودی چی بوده، خوبه.
- یک فایل راهنما (README.txt یا README.md): توش توضیح بدید که برنامه چطور اجرا می شه، پیش نیازها چیه (مثلاً نصب PyQt/NumPy/Pandas در صورت نیاز)، و هر نکته ای که فکر می کنید برای اجرای صحیح لازم داریم. همچنین اگه بخش اختیاری جالبی اضافه کردید، اینجا اشاره کنید تا خدای نکرده موقع تصحیح از قلم نیفته.

اگر نمودار یا گزارش خروجی تولید می کنید (مثلاً فایل FinalGrades.xlsx)، یکی از اون خروجی های نمونه رو هم می تونید ضمیمه کنید. این اجباری نیست چون ما خودمون هم می تونیم خروجی بگیریم، ولی داشتنش ضرری نداره.

مهلت تحویل: تا پایان روز ۱۴ تیر فرصت دارید پروژه رو تحویل بدید. لطفاً زودتر از ددلاین تحویل بدید که اگر مشکلی بود بتونیم رسیدگی کنیم. یادتون باشه این مدت زود می گذره، کار رو به روز آخر موکول نکنید که پروژه ناقص بمونه! تحویل رو در سامانه کوئرا انجام بدید. هر دانشجو یک نسخه تحویل بده.



## پروژه پایانی مبانی برنامه سازی پایتون

امتیاز	بخش ارزیابی
۲۵ نمره	پیاده سازی فاز ۱ (دریافت ورودی ها و ذخیره سازی)
۲۵ نمره	پیاده سازی فاز ۲ (محاسبه میانگین، ماکسیمم و مینیمم)
۲۵ نمره	پیاده سازی فاز ۳ (شمارش قبولی ها و نمایش خروجی)
۲۵ نمره	کیفیت و مستندسازی کد (خوانایی، نام گذاری مناسب متغیرها، توضیحات کافی و ...)
تا ۱۰ نمره	خلاقیت های اختیاری (قابلیت های اضافه بر درخواست پروژه)

### نحوه ی نمره دهی

معیارهای نمره دهی این پروژه در جدول زیر مشخص شده اند. هر فاز بر اساس تحقق وظایف خواسته شده ارزیابی می شود و نیز کیفیت کلی کدنویسی مورد توجه قرار خواهد گرفت. مجموع نمره این پروژه ۱۰۰ است و تا ۱۰ نمره نیز به خاطر پیاده سازی بخش های خلاقانه اختیاری قابل کسب است.

نکته مهم درباره تصحیح خودکار: بخش هایی از پروژه توسط اسکریپت های خودکار تصحیح خواهند شد. یعنی مثلاً ما یک ورودی CSV مشخص رو به برنامه شما می دیم (یا توی فولدرتون می داریم)، بعد برنامه رو اجرا می کنیم یا توابعتون رو صدا می زنیم و خروجی رو با جواب صحیح مقایسه می کنیم. همچنین ممکنه چک کنیم که آیا فایل خروجی مورد انتظار (مثل FinalGrades.xlsx) ساخته شده یا نه و محتوایش درسته یا نه. بنابراین حتماً اسم ها رو مطابق چیزی که گفتیم استفاده کنید (مثلاً اگر قرار شده ستون FinalGrade بسازید دقیقاً همین نام رو بسازید، یا اگر خروجی Excel می دید، حتماً ذکر کنید کجا ذخیره می شه). کوچک بودن/بزرگ بودن حروف در کد و نام فایل ها رو هم مدنظر قرار بدید چون سیستم های خودکار به این چیزا حساسن. نگران نباشید، قبل از تصحیح نهایی ما یکی دو بار برنامه تون رو خودمون هم نگاه می کنیم که نمره بی جهت از دست ندید، اما لطفاً این استانداردها رو رعایت کنید تا کار هم شما هم ما راحت تر بشه. در پایان، اگر به مشکل یا سوالی برخوردید، اصلاً دودل نباشید که بپرسید. ما با کمال میل راهنماییتون می کنیم (البته نه در حد لو دادن جواب آماده! بلکه در حد رفع اشکال یا راهنمایی). هدفمون اینه که از این پروژه چیزی یاد بگیرید و براتون جذاب باشه. مطمئنم وقتی پروژه رو تموم کنید و نتیجه ش رو ببینید، به توانایی های خودتون ایمان میارید و می تونید این تجربه رو در آینده تو پروژه های بزرگ تر به کار ببرید. موفق باشید و منتظریم هنرنمایی شما رو ببینیم!