



# تقرير الأوامر المستخدمة في Git

هذا التقرير يقدم نظرة عامة على الأوامر الأساسية المستخدمة في مشروع هندسة البرمجيات العملي،

مع التركيز على إطار عمل Django.

عمل الطالب : طارق العمري  
إشراف المهندس : مالك المصنف

# إعداد هيكل المشروع

1

إنشاء المجلد الرئيسي

بدءًا من الانتقال إلى المسار المطلوب وإنشاء المجلد **SoftWare-Enginner-**  
**Tareq-Al-Omari** والمجلد الفرعي **Labs**.

```
cd "E:\INFORMATION TECHNOLOGY\learn  
"programing\Projects_Github  
mkdir -p SoftWare-Enginner-Tareq-Al-Omari/Labs
```

2

إنشاء مجلدات المعامل

الانتقال إلى مجلد **Labs** وإنشاء 10 مجلدات فرعية من **Lab1** إلى **Lab10**  
باستخدام حلقة تكرارية.

```
cd SoftWare-Enginner-Tareq-Al-Omari/Labs  
for i in {1..10}; do mkdir "Lab$i"; done
```

# تهيئة المستودع Git

للبدء في تتبع التغييرات وإدارة الإصدارات باستخدام Git، تم تهيئة المستودع في المجلد الرئيسي للمشروع.

```
cd ../../SoftWare-Enginner-Tareq-Al-Omari  
git init
```



# ReadMe

Pocfessional README]

- Cernprse pull ide-taaks forrensuren Treungls(ASma8);  
asdo ussillect(lecalt)  
~\_MMAT);
- Tome : uhatve stpristreerfare vissiorly  
772.
- Lerngreslanfispracres - 1.V1);  
Feddoura ersiorrence arted\_fide in Mesrite bolg\_lit1  
toty  
LyassuIR,,NEPPS =\_README)
- Nakddown whve betal.contacts(55481)  
05(980Ad retadaaattive
- Cas: meacking:  
LestSinckrlally wullficed arents of faperleald):
- Wash(:chausts: and mantbasescration  
Sucaance I -\_Sopaly
- Shoul.cnstemamer((Ienit; orglatren()

## إنشاء ملف README احترافي

تم إنشاء ملف README.md لتقديم وصف شامل للمشروع، مما يسهل على المطورين الآخرين فهمه والتعاون فيه.

```
cat > README.md << 'EOF'
```

# (تمت إضافته هنا README محتوى)

```
EOF
```

# ربط المشروع مع GitHub

لتمكين التعاون ومشاركة المشروع، تم ربط المستودع المحلي بمستودع GitHub البعيد.

```
git remote add origin https://github.com/tareq-alomari/SoftWare-Enginner-Tareq-Al-Omari.git
```



# التتبع والرفع الأولي

بعد إعداد هيكل المشروع، تم تتبع جميع الملفات وإضافتها إلى منطقة التخزين المؤقت، ثم تم الالتزام بالتغييرات ورفعها إلى الفرع الرئيسي على GitHub.

1

إضافة الملفات

```
git add .
```

2

الالتزام بالتغييرات

```
git commit -m "إضافة هيكل واجب Software Engineering مع  
المجلدات من Lab1 إلى Lab10"
```

3

تسمية الفرع الرئيسي

```
git branch -M main
```

4

الرفع إلى GitHub

```
git push -u origin main
```



# إضافة ملفات .gitkeep للمجلدات الفارغة

لضمان ظهور المجلدات الفارغة على GitHub، تم إنشاء ملفات **gitkeep** داخل كل مجلد من مجلدات **Labs**، ثم تم تتبعها ورفعها.

```
for i in {1..10}; do touch Labs/Lab$i/.gitkeep; done  
git add .  
git commit -m "إضافة ملفات .gitkeep على مجلدات Labs"  
git push
```

# العمل مع الفروع

لإدارة الميزات الجديدة أو الإصلاحات بشكل منفصل، تم إنشاء فرع جديد باسم **temp**، ثم تم التبديل إليه، وأخيرًا تم دمج التغييرات من الفرع الرئيسي **main** فيه.

## إنشاء فرع جديد

```
git branch temp
```

## التبديل إلى الفرع الجديد

```
git checkout temp
```

## دمج الفرع الرئيسي

```
git merge main
```



# ملاحظات وأوامر Git إضافية

- تم استخدام **gitkeep** للحفاظ على ظهور مجلدات **Labs** على GitHub.
  - الفرع **main** هو الأساسي، والفرع **temp** تم إنشاؤه للاختبار.
  - المشروع الآن مرفوع بالكامل على GitHub ويحتوي على README وهيكل منظم.
- بالإضافة إلى الأوامر المذكورة، تم استخدام أوامر Git أخرى لإدارة المشروع بفعالية:

```
git config --global user.email "qutabh2@gmail.com"
git config --global user.user "Tareq"
git init
git status
git add 'report.txt'
git commit -m 'I added a report file'
git log
git checkout 'Number'
git branch
git branch 'temp'
git branch 'temp'
git switch 'main'
git switch -c 'temp2'
git switch -b 'temp3'
git merge 'temp'
git ls-files
git clean -dn
git ls-files
git clean -dn
git clean -df
git restore 'report.txt'
git restore .
git rm 'report.txt'
git reset 'temp'
```