1. المقدمة

شهد مجال تطوير البرمجيات وإدارة المشروعات التقنية تطوراً كبيراً مع ظهور أنظمة إدارة الإصدارات (Version Control Systems). يُعد نظام Git أحد أكثر هذه الأنظمة انتشاراً وكفاءة، حيث يتيح للمطورين والباحثين تتبع التغييرات على الملفات والأكواد المصدرية، والرجوع إلى النسخ السابقة، والعمل بمرونة عالية دون التأثير على النسخة الأساسية للمشروع. واكتملت هذه المنظومة بظهور منصة GitHub السحابية، التي توفر بيئة متكاملة لاستضافة المستودعات، والنسخ الاحتياطي، وإدارة الفرق البرمجية، بالإضافة إلى تنظيم عمليات تسليم التكاليف والمشاريع الأكاديمية والعملية.

2. المفهوم والفرق بين Git و GitHub

نظام Git: هو أداة محلية مجانية ومفتوحة المصدر لإدارة الإصدارات، تعمل مباشرة على جهاز المستخدم. تكمن وظيفتها الرئيسية في الاحتفاظ بسجل كامل لجميع التعديلات التي تطرأ على المشروع مع إمكانية التنقل بين المراحل المختلفة للتطوير.

منصة GitHub:هي خدمة سحابية قائمة على الويب تُستخدم لاستضافة مستودعات Git. تتيح للمستخدمين رفع أكوادهم ومشاركتها مع الآخرين عبر الإنترنت، وإدارة طلبات الدمج (Pull Requests)، وتسهيل التعاون بين أفراد الفريق البرمجي في مختلف الأماكن.

3. الخطوات والأوامر التطبيقية للعمل على النظام

تتضمن خطوات العمل والتطبيق العملي لتهيئة المستودع المحلي وربطه بالحساب السحابي التسلسل الآتي:

1. **إعداد بيانات التعريف (Git Configuration)**يتم تحديد هوية المستخدم على النظام المحلي لربط كافة التعديلات القادمة بالاسم البريدي الإلكتروني، وذلك عبر الأمرين:
```
git config --global user.email "example@email.com"
git config --global user.name "username"
```
2. **تهيئة المستودع المحلي (Initialization)**:يتم تحويل المجلد الخاص بالمشروع على الجهاز إلى مستودع تتبع محلي عبر تنفيذ الأمر:
```
git init
```

3. **متابعة حالة الملفات (Check Status):** للتحقق من الملفات المستهدفة والتغيرات غير المسجلة، يتم استخدام الأمر:

```
git status
```

4. **تجهيز الملفات للرفع (Staging Area):** يتم نقل كافة الملفات والتعديلات الحالية إلى منطقة الاستعداد للتثبيت والتسجيل عبر الأمر:

```
git add .
```

5. **تثبيت التعديلات محلياً (Commit):** يتم حفظ التغييرات وتسجيلها رسمياً في سجل التغييرات المحلي مع إضافة وصف توضيحي عبر الأمر:

```
git commit -m "إضافة الملفات والتقرير البرمجي"
```

6. **تعيين الفرع الرئيسي (Branching):** يتم تسمية وتحديد الفرع الأساسي في المستودع بالاسم المعتمد main بواسطة الأمر:

```
git branch -M main
```

7. **ربط المستودع المحلي بالمنصة السحابية (Remote Repository):** يتم ربط مجلد العمل المحلي برابط المستودع الخارجي المنشأ على GitHub باستخدام الأمر:

```
git remote add origin https://github.com/username/repository-name.git
```

8. **رفع الملفات إلى الحساب السحابي (Push):** يتم إرسال ونقل جميع التعديلات والملفات المحفوظة من الجهاز الشخصي إلى منصة GitHub عبر الأمر:

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

9. **نسخ المستودع (Cloning):** يتم استخدام خيار النسخ لتنزيل بيئة مشروع كاملة من منصة GitHub إلى الجهاز المحلي عبر الأمر:

```
git clone <URL>
```

10. **جلب وتحديث البيانات (Pulling):** لضامنة أي تعديلات سحابية جديدة مع البيئة المحلية، يتم استخدام الأمر:

```
git pull
```

4. آلية تنفيذ وتسليم التكليف على GitHub

تتم عملية التسليم الأكاديمي عبر الخطوات العملية الآتية:

- إنشاء نسخة مستودع خاصة (Fork) من المستودع الأصلي المعتمد.
- رفع ملف البحث والتكليف المطلوب بصيغة PDF داخل المستودع المنسوخ عبر زر الرفع (Upload files).
- تأكيد حفظ الملف والتغيرات من خلال إجراء (Commit changes).
- إنشاء وإرسال طلب سحب وتثبيت (Pull Request) لإتاحة التكليف للمراجعة والدمج النهائي في المستودع الرئيسي.