
خطة عمل مشروع نظام التشغيل "NeuronOS" (الاسم المقترح)

الرؤية:

نظام تشغيل ذكي، مخصص لمطوري الذكاء الاصطناعي، يوفر:

- بيئة تطوير وتدريب واختبار متكاملة
- أدوات تحليل وتقييم آلي للمشاريع
- واجهة رسومية مخصصة لاحتياجات المطورين
- تجربة فريدة مدعومة ببعض خصائص الذكاء الاصطناعي داخليًا

المرحلة 1: الإعداد والتحضير (1 يوم)

المهام:

- تنزيل وتشغيل **SerenityOS** (بيئة التطوير)
 - إعداد build system على QEMU أو VirtualBox
 - عمل Fork أو نسخ كاملة لمستودع SerenityOS
 - اختيار اسم وهوية مبدئية للنظام (مثلاً: **NeuronOS**)
 - إعداد GitHub repository خاص بالمشروع
-

🎨 المرحلة 2: تغيير هوية النظام (1.5 يوم)

المهام:

- تغيير الاسم من SerenityOS إلى NeuronOS في جميع الملفات
- تعديل boot screen والشعار والأيقونات الرئيسية
- حذف التطبيقات الغير مفيدة لمطورين الذكاء الاصطناعي (مثل ألعاب أو متصفحات)
- اختيار تصميم مظلم (Dark Theme) مريح للعين أثناء العمل الطويل

🔧 المرحلة 3: تطوير أدوات النظام الأساسية (3 أيام)

الأدوات الأساسية:

1. AI Terminal

- يدعم أوامر طبيعية مثل:

```
train model dataset.csv --epochs 10 --gpu
```

- تكامل مع Python script backend

2. Dataset Browser

- يعرض الملفات بصريًا

- تحليل تنسيقات مثل `.images`, `.json`, `csv`

3. Model Training GUI

- نافذة رسومية لاختيار السكريبت والبيانات والمعلمات

○ تتبع مباشر لحالة التدريب

4. GPU Monitor

○ يعرض استهلاك الموارد (GPU/CPU/RAM)

5. Projects Launcher

○ واجهة تعرض جميع المشاريع النشطة مع روابط مباشرة للبيانات والنماذج

🤖 المرحلة 4: الذكاء الاصطناعي الداخلي (2 يوم)

المميزات:

● مساعد ذكي (AI Agent) داخل الـ Terminal:

○ يرد على أسئلة مثل: "كيف أزيد دقة النموذج؟"

○ مبني باستخدام مكتبة **tinygrad** أو **transformers** لو أمكن

● تقييم المشاريع:

○ عند حفظ مشروع يتم تحليل:

■ جودة البيانات (توازن، نظافة)

■ أداء النموذج (loss/accuracy curves)

■ كفاءة استخدام الموارد

○ ويتم توليد اقتراحات وتحسينات تلقائية

المرحلة 5: واجهة عرض وتحليل النتائج (1.5 يوم)

الأدوات:

- **Graph Visualizer** لعرض:

- Accuracy

- Loss

- Confusion Matrix (إن وجدت)

- تقارير قابلة للحفظ (PDF أو JSON)

المرحلة 6: التوثيق والعرض الخارجي (2 يوم)

التوثيق:

- إعداد صفحة **README.md** على GitHub

- تصوير فيديو يشرح النظام وميزاته

- إنشاء موقع بسيط للمشروع (landing page)

- كتابة مقالة شخصية بعنوان:

"لماذا بنيت نظام تشغيل خاص بالذكاء الاصطناعي؟"

ملخص الخطة والوقت: 

المرحلة	الأيام
الإعداد والتضير	1 يوم
تغيير الهوية والواجهة	1.5 يوم
الأدوات الأساسية	3 أيام
دمج AI والتقييم الآلي	2 يوم
عرض وتحليل النتائج	1.5 يوم
التوثيق والعرض الخارجي	2 يوم
الإجمالي	11 يوم تقريبًا

✂️ الأدوات واللغات المستخدمة:

• C++ / Serenity LibGUI – لتطوير الأدوات والواجهة

• Python – للتدريب والتقييم الآلي

• tinygrad / sklearn / matplotlib

• QEMU – لتجريب النظام

• GitHub Pages – لعرض المشروع

🚀 مميزات المشروع أمام الجامعات:

الميزة التأثير

نظام تشغيل كامل مستوى تقني عالي جدًا

موجه لفئة محددة
فهم دقيق لاحتياج
السوق

تقديم آلي + ذكاء
داخلي
تفكير نقدي وابتكار

عرض احترافي
قابلية عالية للقبول
الجامعي