

# نظام تتبع السعرات الحرارية

## CALORIE TRACKER SYSTEM

### أسماء أعضاء الفريق

2231814721	رواسي محمود الحاج ابراهيم
2231807880	بشينة عبد العظيم الناجح
2221816790	بروج محمد عبدالرحمن

### نظرة عامة على النظام

يهدف هذا النظام إلى تزويد المستخدم بأداة فعالة لتسجيل ومتابعة الاستهلاك اليومي من السعرات الحرارية من خلال واجهة رسومية مبنية بلغة Java. يشمل النظام الميزات الأساسية التالية:

- تسجيل الوجبات: إدخال بيانات الوجبات (الاسم، الكمية، ونوع الوجبة).
- حساب تلقائي: حساب السعرات الحرارية بناءً على الكمية والقيمة الغذائية.
- عرض البيانات: عرض الوجبات المسجلة في جدول محدث.
- تحديد الأهداف: تحديد هدف سعرات يومي ومتابعة التقدم عبر شريط مرئي.
- التخزين والاسترجاع: حفظ البيانات في داتا بيز مع إمكانية العمل دون اتصال.
- تصدير التقارير: حفظ التقرير اليومي كملف نصي.

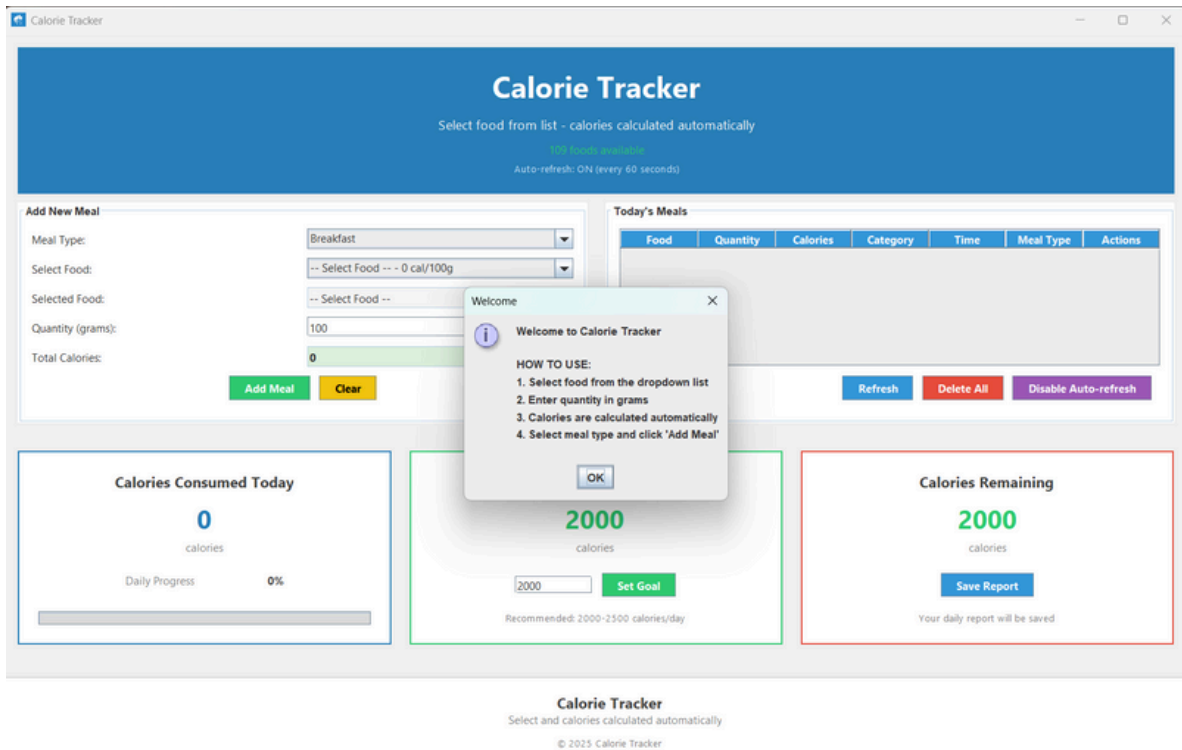
### التقنيات والادوات المستخدمة

- لغة البرمجة: Java
- بيئة التطوير: IntelliJ IDEA
- واجهة رسومية: Java + Swing
- قاعدة البيانات: MySQL و JDBC للاتصال بقاعدة البيانات
- ادارة قاعدة البيانات: DataGrip

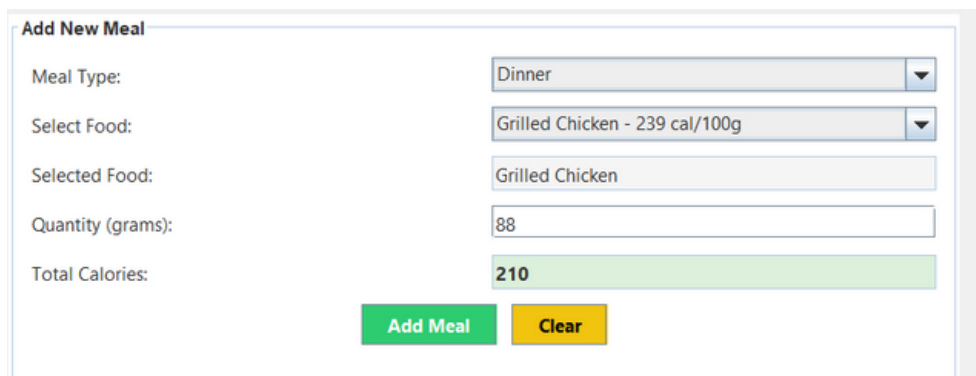
# الميزات الرئيسية المُطبقة

- واجهة Swing متكاملة وتفاعلية.
- حساب سعرات حرارية تلقائي.
- تصنيف وجبات آلي.
- تتبع هدف يومي مع شريط تقدم.
- تخزين في قاعدة بيانات MySQL.
- دعم العمل دون اتصال.
- تصدير تقارير يومية.
- إدارة بيانات كاملة (CRUD).

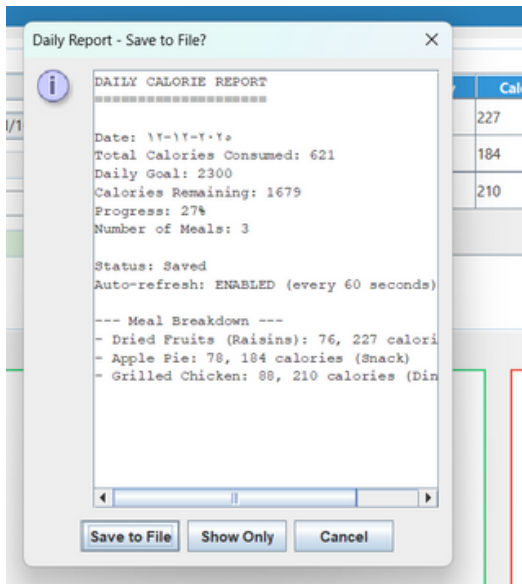
## لقطات شاشة



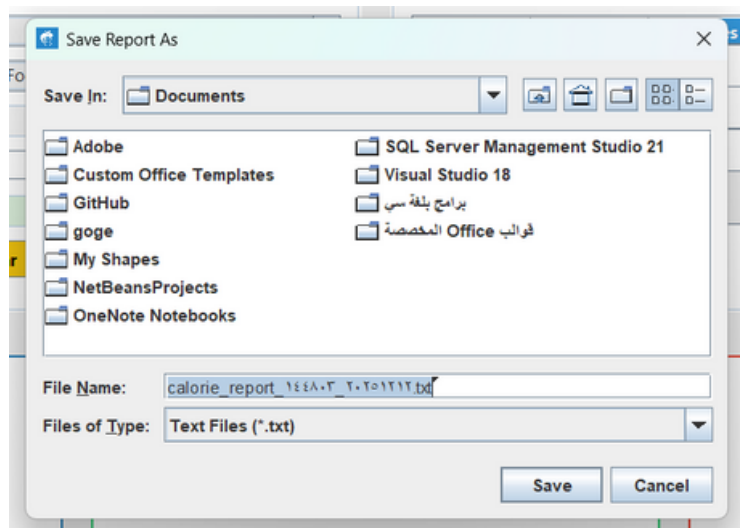
(Opening Page)



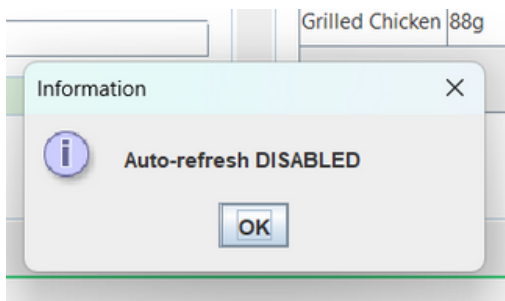
(Adding Meal)



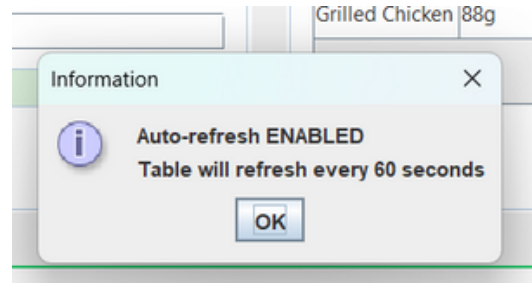
(Daily Report)



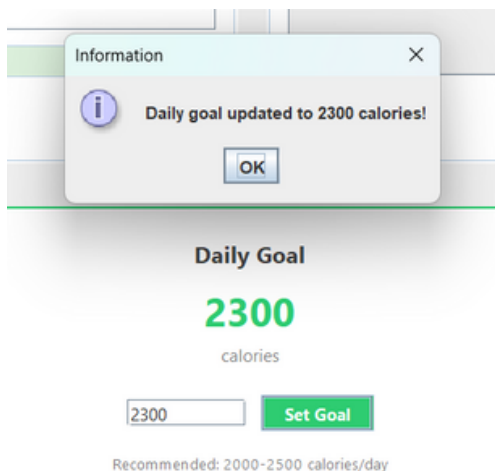
(Saving Daily Report As File)



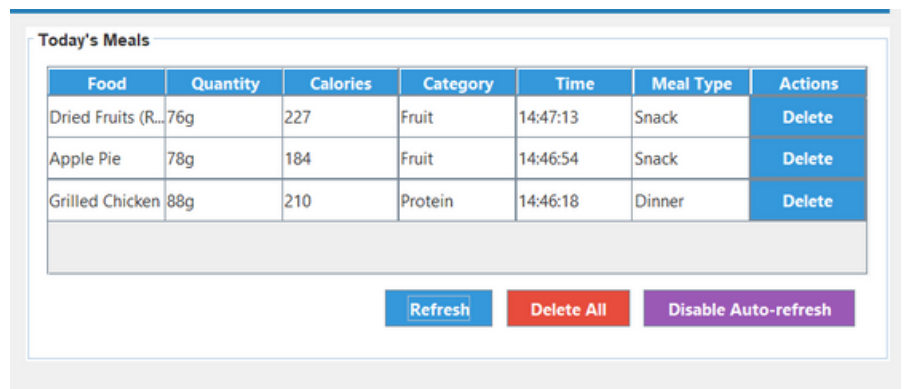
(Enable Auto Refresh)



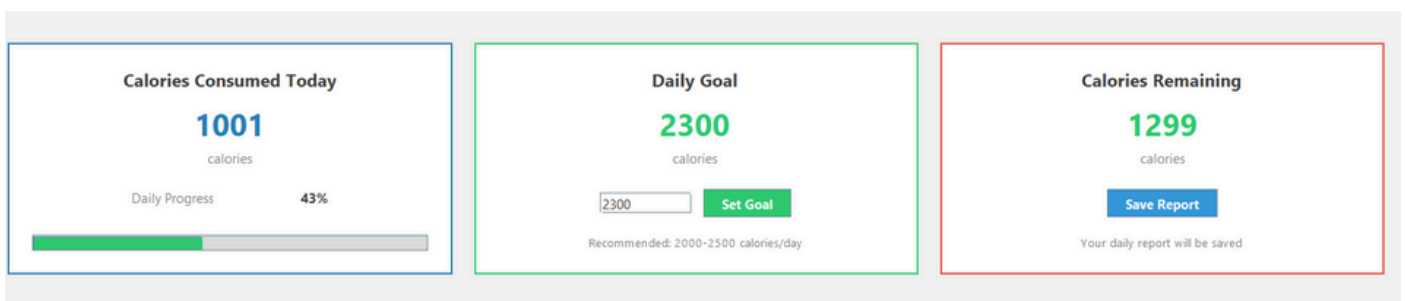
(Disable Auto Refresh)



(Updating Daily Goal)



(User meal Table)



(Progress Bar)

## كيفية تنفيذ المواضيع المطلوبة

العنصر	أهم الأساليب/الأدوات	الهدف
Exception Handling	<ul style="list-style-type: none"> <li>try-catch لخطأ الإدخال والاتصال</li> <li>try-with-resources لإغلاق الموارد</li> </ul>	منع توقف التطبيق ومعالجة الأخطاء بسلاسة
Collections Framework	<ul style="list-style-type: none"> <li>ArrayList&lt;FoodItem&gt; لتخزين الأطعمة</li> <li>List&lt;Meal&gt; لوجبات اليوم</li> </ul>	إدارة بيانات مرنة وسهلة الاسترجاع
JDBC	<ul style="list-style-type: none"> <li>عمليات CRUD</li> <li>PreparedStatement ضد SQL Injection</li> <li>إدارة Connection / ResultSet</li> </ul>	تخزين البيانات بشكل دائم
GUI using Swing	<ul style="list-style-type: none"> <li>JFrame, JPanel, JTable, JComboBox</li> <li>KeyListener و ActionListener للأحداث</li> </ul>	واجهة تفاعلية وسهلة الاستخدام
I/O Streams	<ul style="list-style-type: none"> <li>FileWriter و BufferedWriter لحفظ التقارير ملف و لتسجيل الأحداث</li> <li>JFileChooser لاختيار مكان الحفظ</li> </ul>	تصدير البيانات وتسجيل النشاطات
Multithreading	<ul style="list-style-type: none"> <li>javax.swing.Timer للتحديث التلقائي</li> <li>invokeLater () لتحديث الواجهة بأمان</li> <li>زر تشغيل/إيقاف Auto-refresh</li> </ul>	تحديث البيانات تلقائياً بدون تعطيل الواجهة

## توزيع المهام بين أعضاء الفريق

اسم الطالبة	المهام
رواسي محمود الحاج ابراهيم	<ul style="list-style-type: none"> <li>تصميم جدول عرض الوجبات اليومية</li> <li>تنفيذ أزرار الإجراءات داخل الجدول</li> <li>تطبيق شريط التقدم والإحصائيات</li> <li>إنشاء نظام تحديث البيانات تلقائياً</li> <li>معالجة أحداث الحذف والتعديل</li> </ul>

<ul style="list-style-type: none"> <li>• تصميم الواجهة الرئيسية وتخطيطها</li> <li>• إنشاء الألوان والخطوط والتنسيق</li> <li>• تطبيق نماذج إدخال البيانات</li> <li>• تصميم أقسام Header &amp; Footer</li> <li>• تنفيذ مربعات الاختيار والحقول النصية</li> </ul>	<p>بشينة عبد العظيم الناجح</p>
<ul style="list-style-type: none"> <li>• ربط التطبيق مع قاعدة البيانات MySQL</li> <li>• تنفيذ عمليات حفظ وجلب البيانات</li> <li>• نظام التحديث التلقائي متعدد الخيوط</li> <li>• تطبيق نظام حفظ التقارير في الملفات</li> <li>• برمجة المنطق الأساسي للتطبيق</li> </ul>	<p>بروج محمد عبد الرحمن</p>

## التحديات

### 1. تكامل واجهة Swing مع قاعدة البيانات

- صعوبة مزامنة تحديث الواجهة مع تغييرات البيانات
- حاجة إلى آليات لتحديث الجدول تلقائياً

### 2. إدارة الاتصال بقاعدة البيانات

- معالجة فشل الاتصال بشكل أنيق
- التحول بين وضع الاتصال والوضع المحلي

### 3. تزامن الخيوط في البرمجة

- ضمان تحديث الواجهة من الخيوط الخلفية بأمان
- منع تعارض تحديثات الجداول

### 4. تصميم واجهة مستخدم معقدة

- تنظيم عدد كبير من المكونات في تخطيط متناسق
- تحقيق تجربة مستخدم سلسة وتلقائية

### 5. تحديات العمل الجماعي

#### تنسيق العمل على ملفات مشتركة:

- تجنب التعارض في تعديلات الكود
- ضمان التوافق بين أجزاء المشروع المختلفة

#### توحيد نمط التصميم والبرمجة:

- الحفاظ على تناسق الألوان والخطوط عبر الواجهة
- اتباع نفس نمط كتابة الكود

#### اختبار التكامل:

- اكتشاف الأخطاء الناتجة عن تفاعل الأجزاء المختلفة
- التأكد من عمل جميع الميزات معاً

# الدروس التقنية المستفادة

## 1. تصميم واجهات Swing متقدمة

- فهم عميق لتخطيطات المختلفة (**GridLayout** و **BorderLayout**)
- خبرة في استخدام مكونات **Swing** المتقدمة

## 2. إدارة قواعد البيانات في Java

- التعامل مع استعلامات **SQL** وعمليات **CRUD**
- معالجة استثناءات الاتصال بقواعد البيانات

## 3. معالجة الملفات والنصوص

- استخدام **JFileChooser** لاختيار الملفات
- كتابة التقارير النصية وتنسيقها