

نظام تتبع السعرات الحرارية

CALORIE TRACKER SYSTEM

أسماء أعضاء الفريق

2231814721	رواسي محمود الحاج ابراهيم
2231807880	بثينة 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، نوع 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 دата بيز مع إمكانية العمل دون اتصال.
- تصدير التقارير: حفظ التقرير اليومي كملف نصي.

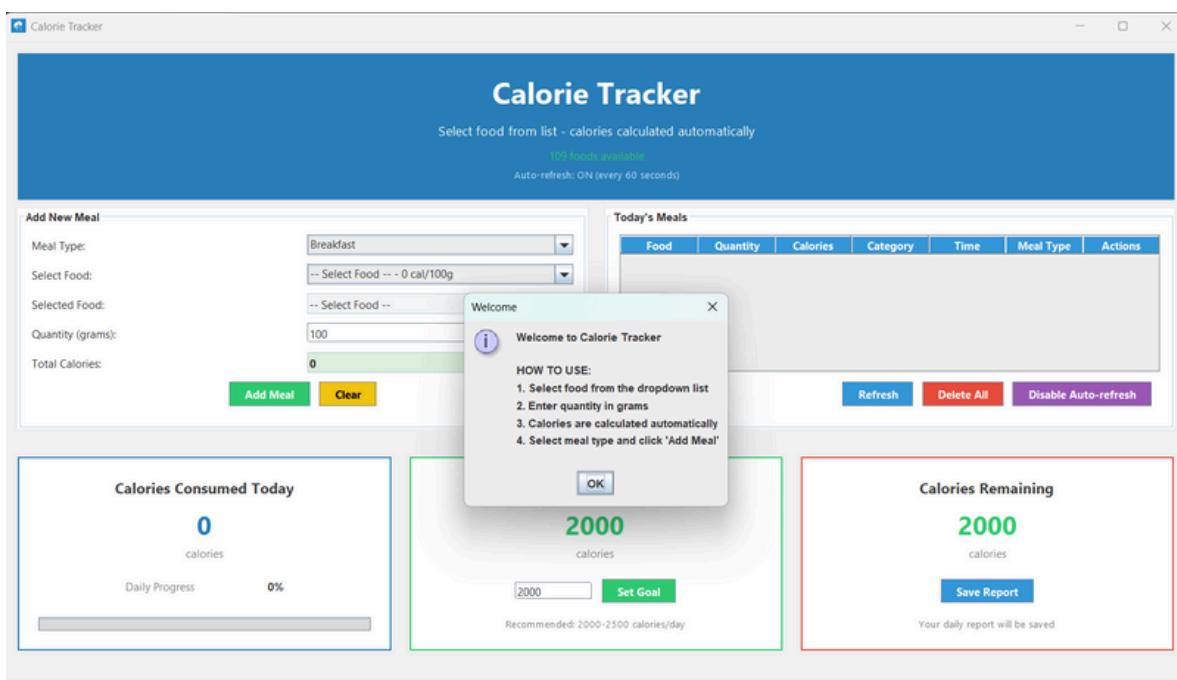
التقنيات والادوات المستخدمة

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

الميزات الرئيسية المطبقة

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

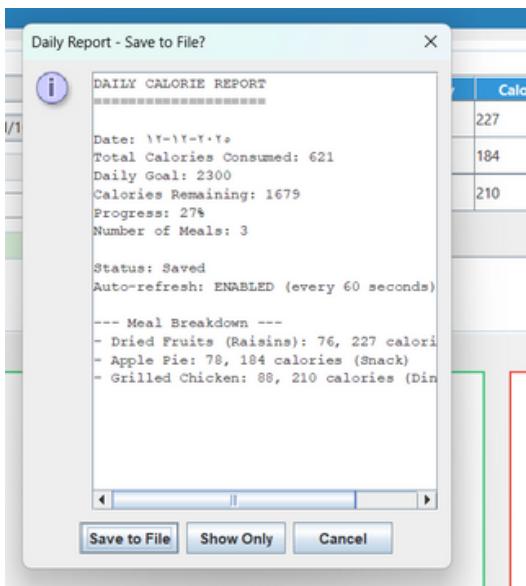
لقطات شاشة



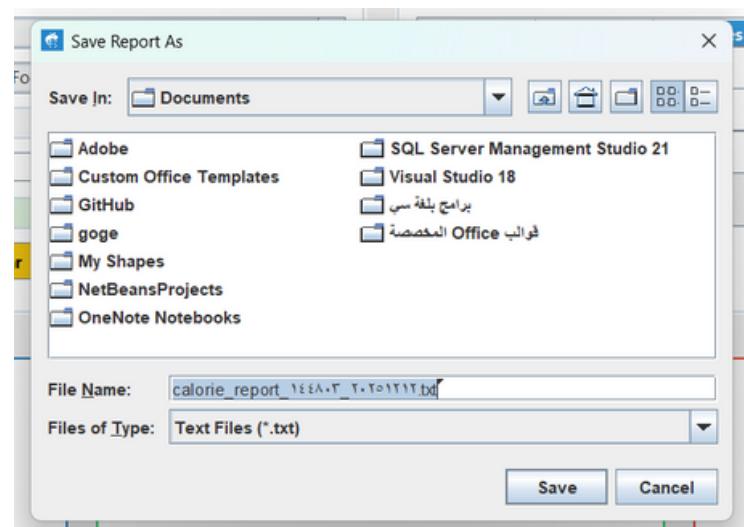
(Opening Page)

A screenshot of the "Add New Meal" form. It has fields for Meal Type (Dinner), Select Food (Grilled Chicken - 239 cal/100g), Selected Food (Grilled Chicken), Quantity (grams) (88), and Total Calories (210). There are "Add Meal" and "Clear" buttons at the bottom.

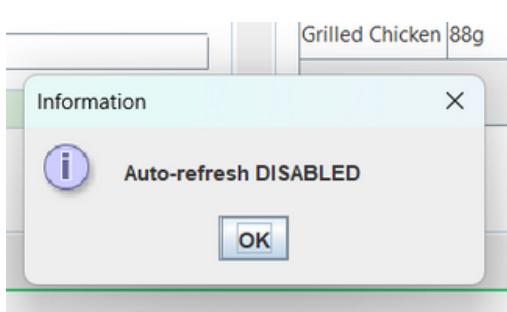
(Adding Meal)



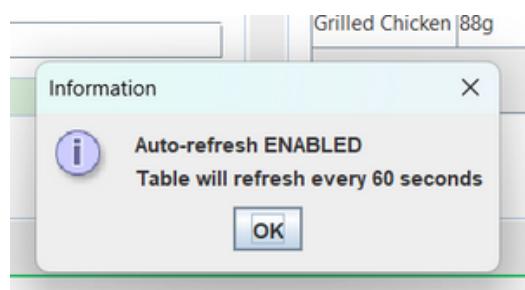
(Daily Report)



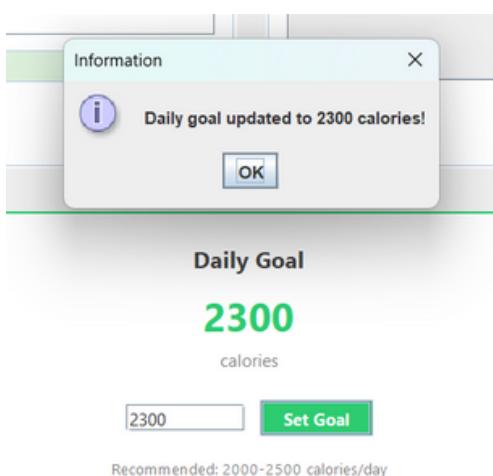
(Saving Daily Report As File)



(Enable Auto Refresh)



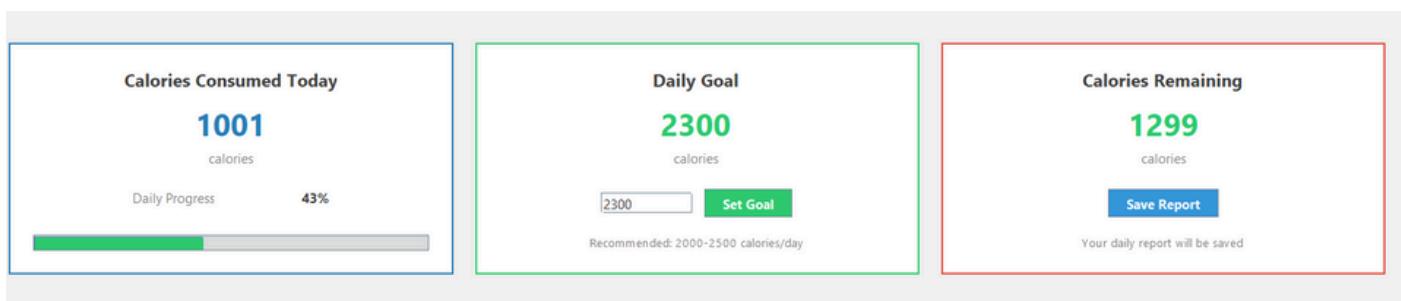
(Disable Auto Refresh)



(Updating Daily Goal)

Today's Meals						
Food	Quantity	Calories	Category	Time	Meal Type	Actions
Dried Fruits (R...	76g	227	Fruit	14:47:13	Snack	Delete
Apple Pie	78g	184	Fruit	14:46:54	Snack	Delete
Grilled Chicken	88g	210	Protein	14:46:18	Dinner	Delete

(User meal Table)



(Progress Bar)

كيفية تنفيذ المواضيع المطلوبة

الهدف	أهم الأساليب/الأدوات	العنصر
منع توقف التطبيق ومعالجة الأخطاء بسلسلة	لأخطاء الإدخال والاتصال try-catch • لإغلاق الموارد try-with-resources •	Exception Handling
إدارة بيانات مرنة وسهلة الاسترجاع	ArrayList<FoodItem> لتخزين الأطعمة List<Meal> لوجبات اليوم •	Collections Framework
تخزين البيانات بشكل دائم	عمليات CRUD • SQL ضد PreparedStatement • Injection • Connection / ResultSet • إدارة	JDBC
واجهة تفاعلية وسهلة الاستخدام	JFrame, JPanel, JTable, JComboBox KeyListener و ActionListener • للأحداث	GUI using Swing
تصدير البيانات وتسجيل النشاطات	FileWriter و BufferedWriter لحفظ التقارير ملف .log لتسجيل الأحداث JFileChooser لاختيار مكان الحفظ •	I/O Streams
تحديث البيانات تلقائياً بدون تعطيل الواجهة	javax.swing.Timer للتحديث التلقائي invokeLater () لتحديث الواجهة بأمان زر تشغيل/إيقاف Auto-refresh •	Multithreading

توزيع المهام بين أعضاء الفريق

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

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

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 تحداثيات 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 لتنظيمات 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