

Basic Application Development Using Python



NAME: MD. Shajib Sikder

Technology: Computer Science

Roll: 862768 **Reg. No:** 1502357345

Semester: 3rd **Session:** 2023-2024

Project Title: Converter Tool

Project Description: এই প্রজেক্টটি একটি Python Tkinter ভিত্তিক ডেস্কটপ কনভার্টার অ্যাপ যা ইংরেজি কোনো বাক্য, শব্দ বা সংখ্যা-কে Binary, Octal, Decimal, এবং Hexadecimal ফর্মেটে রূপান্তর করতে পারে। প্রজেক্টটি পুরোপুরি অফলাইন ভিত্তিক এবং কোনো ইন্টারনেট পারমিশন প্রয়োজন হয় না।

🔍 প্রজেক্টের প্রধান বৈশিষ্ট্যসমূহ:

- 🎬 আকর্ষণীয় Splash Screen
- 📱 মোবাইল-সাইজ UI Layout
- 🗂️ Text বা Number কে চারটি ডিজিটাল ফর্ম্যাটে রূপান্তর
- 📖 How to Work নির্দেশনা সেকশন
- 🖼️ (অপশনাল) ইমেজ স্লাইডার ফিচার
- 📊 রূপান্তরের ধাপে ধাপে ব্যাখ্যা সহ ফলাফল পেজ
- ⬅️ BACK Back এবং 🚪 Exit অপশন
- 💻 পুরো অ্যাপটি PyCharm IDE-তে রান করা যায়

স্প্যাশ স্ক্রিন (Splash Screen)

স্প্যাশ স্ক্রিন (Splash Screen): অ্যাপ্লিকেশন চালু হওয়ার সময় এটি প্রথম প্রদর্শিত হয়। এটি একটি স্বাগত বার্তা এবং লোডিং অ্যানিমেশন দেখায়।

- বর্তমান ফ্রেমটি মুছে ফেলে একটি নতুন স্প্যাশ ফ্রেম তৈরি করে।
- "CONVERTER TOOL" শিরোনাম এবং "Loading..." টেক্সট প্রদর্শন করে।
- update_loading_text ফাংশনটি লোডিং টেক্সটের ডট অ্যানিমেশন নিয়ন্ত্রণ করে।
- self.root.after(5000, self.show_home_page) ব্যবহার করে ৫ সেকেন্ড পর স্বয়ংক্রিয়ভাবে হোম পেজে চলে যায়।

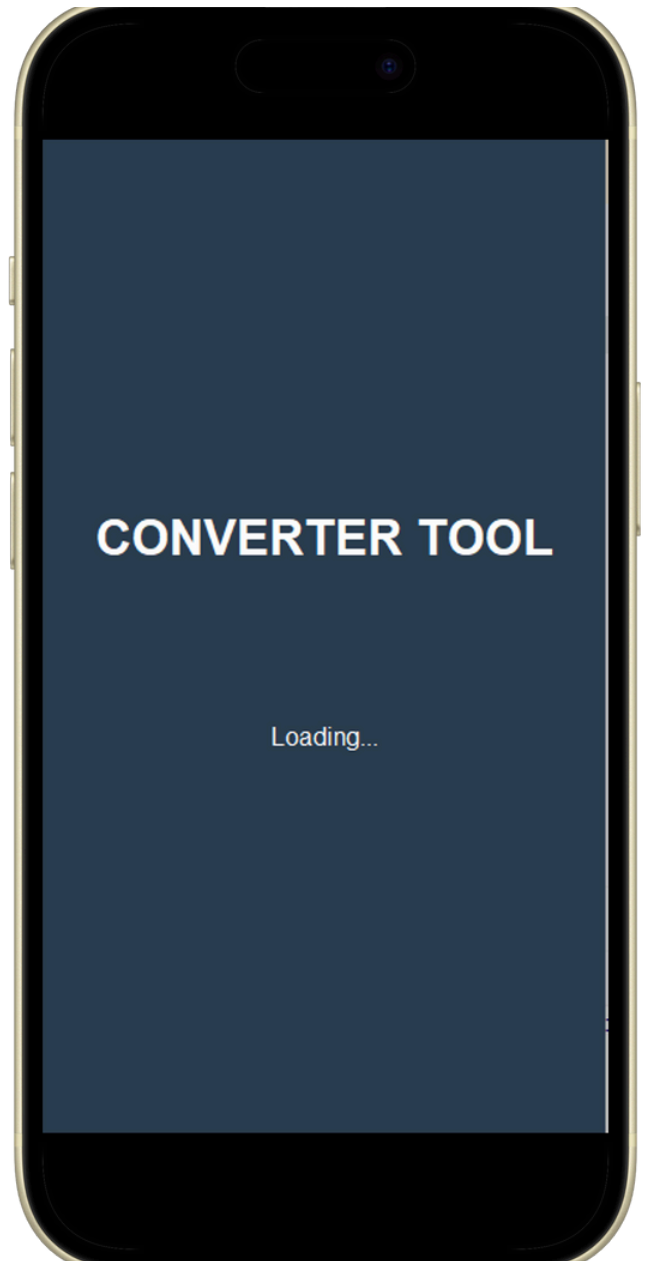
```
def show_splash_screen(self):
    """স্প্যাশ স্ক্রিন প্রদর্শন করে।"""
    self.clear_frame()
    splash_frame = tk.Frame(self.root, bg="#2c3e50")
    # গাঢ় নীল/ধূসর ব্যাকগ্রাউন্ড
    splash_frame.pack(fill="both", expand=True)
    self.current_frame = splash_frame

    # "CONVERTER TOOL" লেখা
    title_label = tk.Label(splash_frame,
        text="CONVERTER TOOL", font=self.title_font,
        fg="white", bg="#2c3e50")
    title_label.place(relx=0.5, rely=0.4,
        anchor="center")

    # লোডিং টেক্সট
    self.loading_label = tk.Label(splash_frame,
        text="Loading...", font=self.text_font, fg="white",
        bg="#2c3e50")
    self.loading_label.place(relx=0.5, rely=0.6,
        anchor="center")

    # লোডিং অ্যানিমেশন (৫ সেকেন্ড)
    self.loading_dots = 0
    self.update_loading_text()
    self.root.after(5000, self.show_home_page) # ৫
    সেকেন্ড পর হোম পেজে যাবে

def update_loading_text(self):
    """লোডিং টেক্সট অ্যানিমেশন আপডেট করে।"""
    dots = "." * (self.loading_dots % 4)
```



হোম পেজ (Home Page)

এটি অ্যাপ্লিকেশনটির প্রধান ইন্টারফেস, যেখানে ব্যবহারকারী ইনপুট দিতে এবং রূপান্তরের ধরন (বাইনারি, অক্টাল, ডেসিমাল, হেক্সাডেসিমাল) নির্বাচন করতে পারে।

- বর্তমান ফ্রেমটি মুছে ফেলে একটি নতুন হোম ফ্রেম তৈরি করে।
- অ্যাপ্লিকেশনের শিরোনাম, ইনপুট ফিল্ড এবং চারটি রূপান্তর বাটন (বাইনারি, অক্টাল, ডেসিমাল, হেক্সাডেসিমাল) প্রদর্শন করে।
- "How to Use This Tool" অংশে ব্যবহার নির্দেশিকা দেখায়।
- অ্যাপ্লিকেশন বন্ধ করার জন্য "Exit " বাটনটি যোগ করে।
- create_conversion_button ফাংশনটি প্রতিটি রূপান্তর বাটন তৈরি করতে সাহায্য করে।

```
def show_home_page(self):
    """হোম পেজ প্রদর্শন করে।"""
    self.clear_frame()
    home_frame = tk.Frame(self.root, bg="#ecf0f1") #
    হালকা ধূসর ব্যাকগ্রাউন্ড
    home_frame.pack(fill="both", expand=True)
    self.current_frame = home_frame

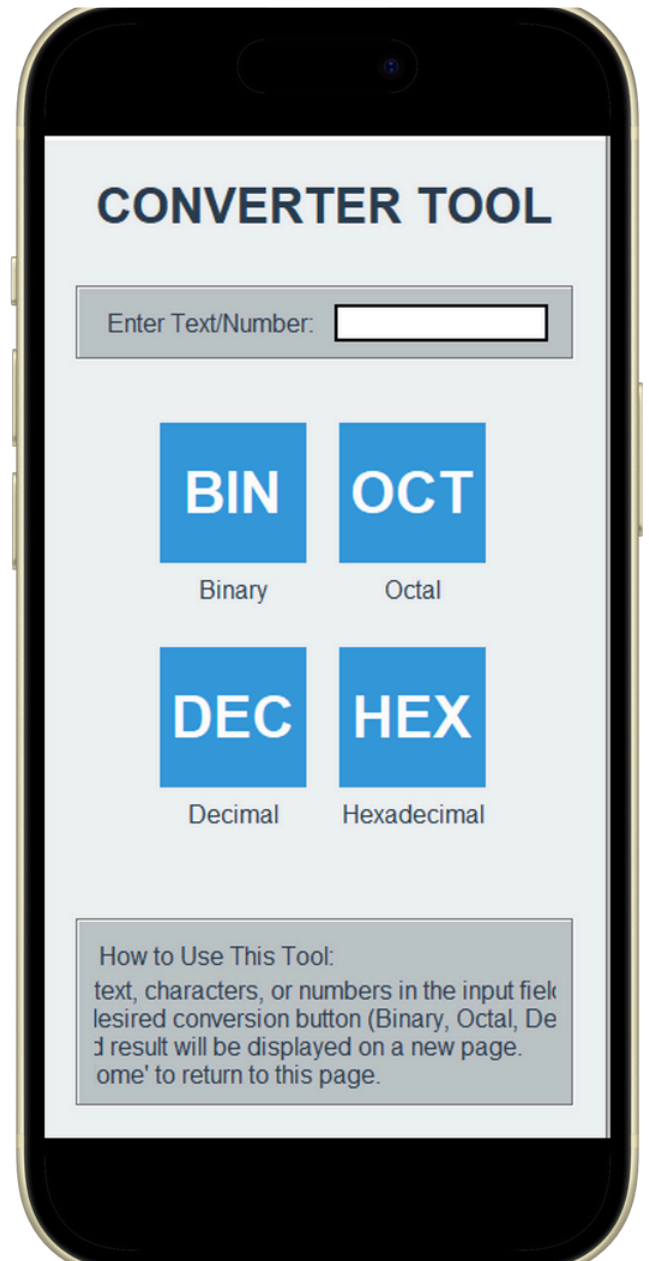
    # "CONVERTER TOOL" শিরোনাম
    title_label = tk.Label(home_frame,
        text="CONVERTER TOOL", font=self.title_font,
        fg="#2c3e50", bg="#ecf0f1")
    title_label.pack(pady=20)

    # ইনপুট ফিল্ড
    input_frame = tk.Frame(home_frame,
        bg="#bdc3c7", padx=10, pady=10, relief="groove",
        bd=2)
    input_frame.pack(pady=10, padx=20, fill="x")

    input_label = tk.Label(input_frame, text="Enter
    Text/Number:", font=self.text_font, fg="#2c3e50",
    bg="#bdc3c7")
    input_label.pack(side="left", padx=5)

    self.input_entry = tk.Entry(input_frame,
    font=self.text_font, width=25, bd=2, relief="solid")
    self.input_entry.pack(side="right", expand=True,
    fill="x", padx=5)

    # এক্সিট বাটন
    exit_button = tk.Button(home_frame, text="Exit ",
    font=self.text_font, command=self.root.destroy,
    bg="#e74c3c", fg="white",
    activebackground="#c0392b", relief="raised", bd=2)
    exit_button.pack(pady=10)
```

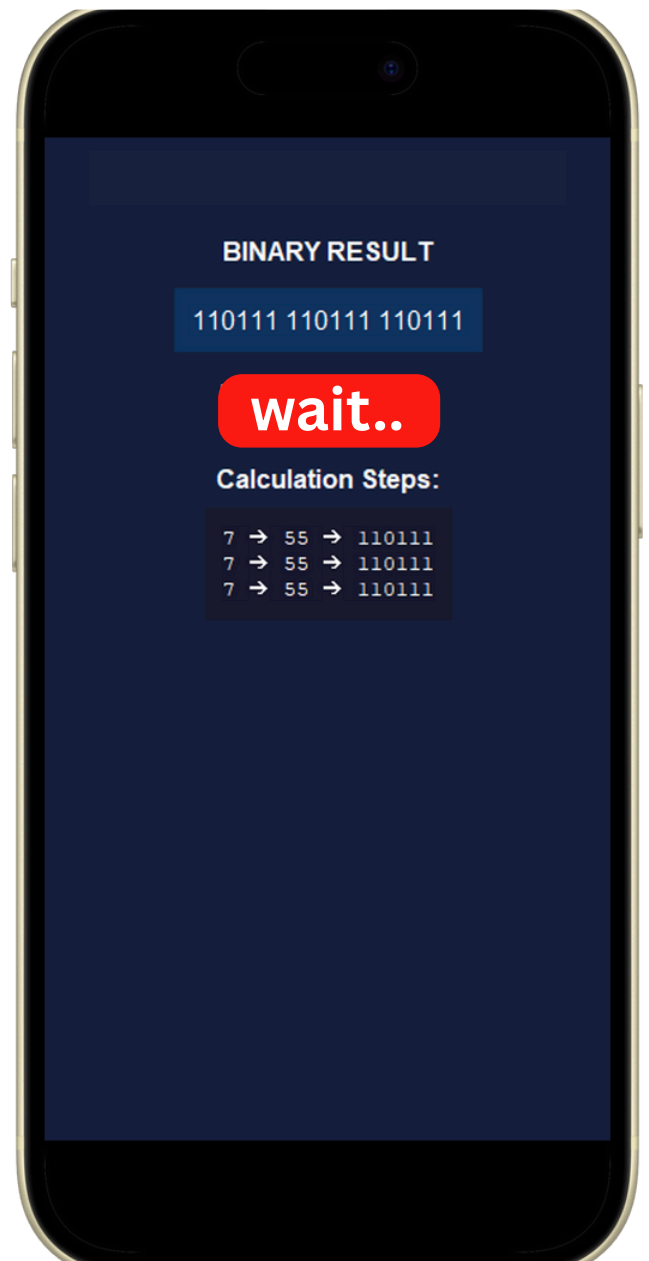


রূপান্তর লজিক (Conversion Logic)

এই অংশটি ইনপুট ডেটাকে নির্বাচিত ফরম্যাটে রূপান্তর করে এবং ধাপে ধাপে ব্যাখ্যার জন্য ডেটা তৈরি করে।

- `perform_conversion` ফাংশনটি ইনপুট ফিল্ড থেকে ডেটা নেয়।
- প্রতিটি অক্ষরকে তার ASCII/Unicode ডেসিমাল মানের রূপান্তর করে।
- নির্বাচিত রূপান্তরের ধরন অনুযায়ী (বাইনারি, অক্টাল, ডেসিমাল, হেক্সাডেসিমাল) সেই ডেসিমাল মানকে রূপান্তরিত করে।
- `_get_decimal_to_base_explanation` ফাংশনটি ব্যবহার করে প্রতিটি রূপান্তরের ধাপে ধাপে গাণিতিক ব্যাখ্যা তৈরি করে।
- রূপান্তরিত ফলাফল এবং ব্যাখ্যা সহ `show_results_page` ফাংশনকে কল করে।

```
def _get_decimal_to_base_explanation(self, decimal_val, base, base_name):  
    """  
    একটি ডেসিমাল মানকে নির্দিষ্ট বেসে রূপান্তর করার ধাপে  
    ধাপে ব্যাখ্যা তৈরি করে।  
    """  
    explanation_lines = []  
    explanation_lines.append(f" {decimal_val}  
(ডেসিমাল) কে {base_name} এ রূপান্তর:")  
    # ... (ভাগশেষ পদ্ধতির লজিক) ...  
    return "\n".join(explanation_lines)  
  
def perform_conversion(self, conversion_type):  
    """ইনপুট ডেটা নির্দিষ্ট ফরম্যাটে রূপান্তর করে এবং ফলাফল  
    পেজ দেখায়।"""  
    input_data = self.input_entry.get()  
    if not input_data:  
        self.show_message_box("Input Error", "অনুগ্রহ  
করে রূপান্তর করার জন্য কিছু টেক্সট বা সংখ্যা লিখুন।")  
        return  
    converted_result_parts.append(bin(decimal_val)  
[2:])  
  
    explanation_parts.append(self._get_decimal_to_base_e  
xplanation(decimal_val, 2, "বাইনারি"))  
    # ... (অক্টাল, ডেসিমাল, হেক্সাডেসিমাল লজিক) ...  
    explanation_parts.append("\n")  
  
    converted_result = " ".join(converted_result_parts)  
    explanation = "\n".join(explanation_parts).strip()  
  
    self.show_results_page(input_data,  
conversion_type.capitalize(), converted_result,  
explanation)
```



ফলাফল পেজ (Result Page)

এই পেজটি রূপান্তরিত ফলাফল এবং রূপান্তরের বিস্তারিত ব্যাখ্যা প্রদর্শন করে।

- বর্তমান ফ্রেমটি মুছে ফেলে একটি নতুন ফলাফল ফ্রেম তৈরি করে।
- মূল ইনপুট, রূপান্তরিত ফলাফল এবং tk.Text উইজেটে ধাপে ধাপে রূপান্তরের ব্যাখ্যা প্রদর্শন করে।
- tk.Text উইজেটটি শুধুমাত্র পড়ার জন্য সেট করা হয়।
- হোম পেজে ফিরে যাওয়ার জন্য "Back Home " বাটনটি যোগ করে।

```
def show_results_page(self, original_input,
conversion_type, result, explanation):
    """ফলাফল পেজ প্রদর্শন করে।"""
    self.clear_frame()
    results_frame = tk.Frame(self.root, bg="#ecf0f1")
    # হালকা ধূসর ব্যাকগ্রাউন্ড
    results_frame.pack(fill="both", expand=True)
    self.current_frame = results_frame

    # "CONVERTER TOOL" শিরোনাম
    title_label = tk.Label(results_frame,
text="CONVERTER TOOL", font=self.title_font,
fg="#2c3e50", bg="#ecf0f1")
    title_label.pack(pady=20)

    # ফলাফল বাক্স
    result_box_frame = tk.Frame(results_frame,
bg="#bdc3c7", padx=15, pady=15, relief="groove",
bd=2)
    result_box_frame.pack(pady=10, padx=20,
fill="both", expand=True)
    # "কিভাবে রূপান্তর করা হলো" বাক্স
    explanation_label = tk.Label(result_box_frame,
text="কিভাবে রূপান্তর করা হলো:", font=self.text_font,
fg="#2c3e50", bg="#bdc3c7", anchor="w")
    explanation_label.pack(fill="x", pady=5)
    # "Back Home" বাটন
    back_button = tk.Button(results_frame,
text="Back Home ", font=self.button_font,
command=self.show_home_page, bg="#3498db",
fg="white", activebackground="#2980b9",
relief="raised", bd=2)
    back_button.pack(pady=10)
```



কনভার্টার টুলের ভবিষ্যৎ উন্নতির ধারণা

আপনার বর্তমান "কনভার্টার টুল"টি বেশ কার্যকর। এটিকে আরও উন্নত এবং ব্যবহারকারী-বান্ধব করার জন্য এখানে কিছু ধারণা দেওয়া হলো:

- বিপরীত রূপান্তর (Reverse Conversion): * ধারণা: বর্তমানে আপনি টেক্সট/সংখ্যাকে বাইনারি/অক্টাল/ডেসিমাল/হেক্সাডেসিমালে রূপান্তর করতে পারেন। ভবিষ্যতে বাইনারি, অক্টাল, হেক্সাডেসিমাল বা ডেসিমাল সংখ্যাকে আবার মূল টেক্সট বা সংখ্যায় রূপান্তর করার একটি বিকল্প যোগ করা যেতে পারে। * উপকারিতা: এটি টুলটিকে আরও বহুমুখী করবে এবং ব্যবহারকারীদের উভয় দিকেই রূপান্তর করার সুবিধা দেবে।
- ঐতিহাসিক লগ (History Log): * ধারণা: ব্যবহারকারী যে রূপান্তরগুলো করেছে, সেগুলোর একটি তালিকা বা ইতিহাস সংরক্ষণ করা যেতে পারে। এটি একটি ছোট টেক্সট ফাইল বা অ্যাপ্লিকেশনের ভেতরেই একটি স্ক্রলযোগ্য তালিকা হিসেবে দেখানো যেতে পারে। * উপকারিতা: ব্যবহারকারীরা তাদের পূর্ববর্তী রূপান্তরগুলো দ্রুত দেখতে এবং প্রয়োজনে আবার ব্যবহার করতে পারবে, যা সময় বাঁচাবে।
- ক্লিপবোর্ডে কপি করার সুবিধা (Copy to Clipboard): * ধারণা: রূপান্তরিত ফলাফল এবং ব্যাখ্যার টেক্সট ক্লিপবোর্ডে কপি করার জন্য একটি বাটন যোগ করা যেতে পারে। * উপকারিতা: ব্যবহারকারীরা সহজেই ফলাফল অন্য কোথাও পেস্ট করতে পারবে, যা তাদের কাজের প্রবাহকে মসৃণ করবে।
- থিম পরিবর্তন (Theme Customization): * ধারণা: ব্যবহারকারীদের জন্য অ্যাপ্লিকেশনের থিম বা কালার স্কিম পরিবর্তন করার একটি বিকল্প যোগ করা যেতে পারে (যেমন, লাইট মোড, ডার্ক মোড বা কাস্টম কালার)। * উপকারিতা: এটি ব্যবহারকারীর ব্যক্তিগত পছন্দ অনুযায়ী অ্যাপ্লিকেশনের চেহারা পরিবর্তন করার সুযোগ দেবে এবং ব্যবহারকারীর অভিজ্ঞতা উন্নত করবে।
- ইনপুট ক্লিয়ার বাটন (Clear Input Button): * ধারণা: ইনপুট ফিল্ডের পাশে একটি ছোট "Clear" বাটন যোগ করা যেতে পারে, যা ক্লিক করলে ইনপুট ফিল্ডটি খালি হয়ে যাবে। * উপকারিতা: ব্যবহারকারীকে ম্যানুয়ালি ইনপুট মুছে ফেলার কামেলা থেকে মুক্তি দেবে এবং নতুন ইনপুট দেওয়ার প্রক্রিয়াকে দ্রুত করবে।