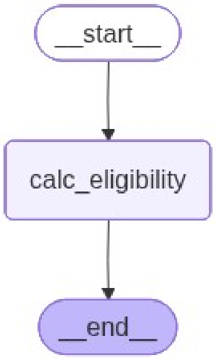
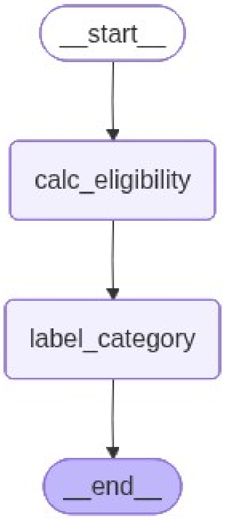
LangGraph Sequential flow

Lets say you want to automate a standard process, instead of using Agents, they will take over the entire stuff, we can have a workflow and decide how this will go. By doing this, we are having a control on how it will start and end but in between operations will involve AI. In LangGraph, we specify the entire flow, how it will start and end. We are using the AI power only in the functions in between. Lets say we have a workflow, which has 5 steps, from start to end (we can have a fixed starting point and ending point), in between whatever step you do that will involve AI.

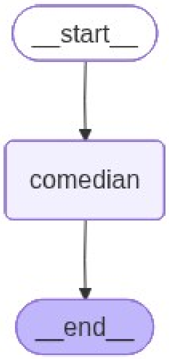
Ran loan_eligibility.py to generate png



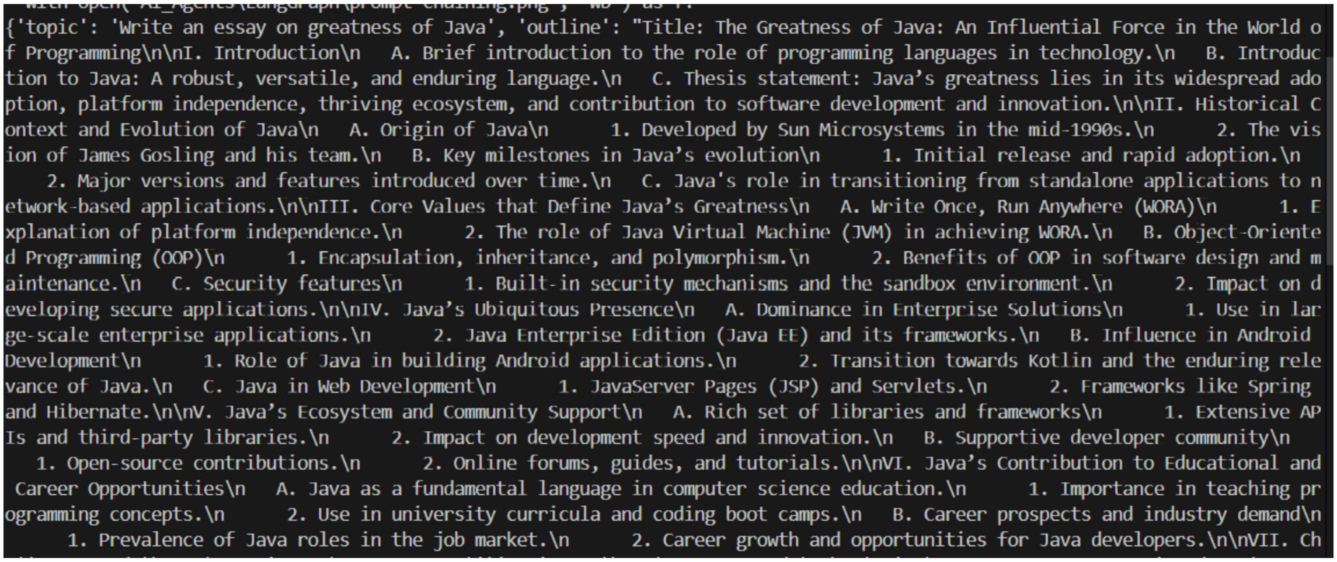
Ran loan_eligibility_cat.py



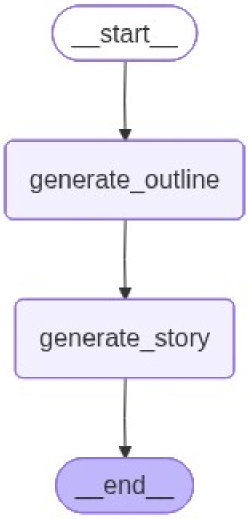
Ran AI_comedian_workflow



Ran prompt-chaining.py



```
  with open('AI_Agents\LangGraph\prompt-chaining.png', 'wb') as f:
{'topic': 'Write an essay on greatness of Java', 'outline': "Title: The Greatness of Java: An Influential Force in the World o
f Programming\n\nI. Introduction\n   A. Brief introduction to the role of programming languages in technology.\n   B. Introduc
tion to Java: A robust, versatile, and enduring language.\n   C. Thesis statement: Java’s greatness lies in its widespread ado
ption, platform independence, thriving ecosystem, and contribution to software development and innovation.\n\nII. Historical C
ontext and Evolution of Java\n   A. Origin of Java\n      1. Developed by Sun Microsystems in the mid-1990s.\n      2. The vis
ion of James Gosling and his team.\n   B. Key milestones in Java’s evolution\n      1. Initial release and rapid adoption.\n
    2. Major versions and features introduced over time.\n   C. Java's role in transitioning from standalone applications to n
etwork-based applications.\n\nIII. Core Values that Define Java’s Greatness\n   A. Write Once, Run Anywhere (WORA)\n      1. E
xplanation of platform independence.\n      2. The role of Java Virtual Machine (JVM) in achieving WORA.\n   B. Object-Oriente
d Programming (OOP)\n      1. Encapsulation, inheritance, and polymorphism.\n      2. Benefits of OOP in software design and m
aintenance.\n   C. Security features\n      1. Built-in security mechanisms and the sandbox environment.\n      2. Impact on d
eveloping secure applications.\n\nIV. Java’s Ubiquitous Presence\n   A. Dominance in Enterprise Solutions\n      1. Use in lar
ge-scale enterprise applications.\n      2. Java Enterprise Edition (Java EE) and its frameworks.\n   B. Influence in Android
Development\n      1. Role of Java in building Android applications.\n      2. Transition towards Kotlin and the enduring rele
vance of Java.\n   C. Java in Web Development\n      1. JavaServer Pages (JSP) and Servlets.\n      2. Frameworks like Spring
and Hibernate.\n\nV. Java’s Ecosystem and Community Support\n   A. Rich set of libraries and frameworks\n      1. Extensive AP
Is and third-party libraries.\n      2. Impact on development speed and innovation.\n   B. Supportive developer community\n
   1. Open-source contributions.\n      2. Online forums, guides, and tutorials.\n\nVI. Java’s Contribution to Educational and
 Career Opportunities\n   A. Java as a fundamental language in computer science education.\n      1. Importance in teaching pr
ogramming concepts.\n      2. Use in university curricula and coding boot camps.\n   B. Career prospects and industry demand\n
      1. Prevalence of Java roles in the job market.\n      2. Career growth and opportunities for Java developers.\n\nVII. Ch
```



Output:

```
{'topic': 'Write an essay on greatness of Java', 'outline': 'Title: The Greatness of Java: The Time-Tested
Programming Language\n\nI. Introduction\n   A. Brief overview of Java\'s history\n      1. Origin in the
mid-1990s by Sun Microsystems\n      2. Java\'s growth and evolution\n   B. Importance of
programming languages in the tech industry\n   C. Thesis statement: Java\'s greatness stems from its
platform independence, robust ecosystem, versatility, and continuous adaptability.\n\nII. Platform
Independence\n   A. Explanation of "Write Once, Run Anywhere" (WORA)\n   B. Java Virtual Machine
(JVM) as the key enabler\n   C. Portability on diverse hardware and operating systems\n   D. Case
studies of applications utilizing Java’s platform-independent nature\n\nIII. Robustness and Security\n
A. Memory management and exception handling\n      1. Automatic garbage collection\n      2. Strong
type checking\n   B. Security features\n      1. Bytecode verification\n      2. Security API and access
control functionalities\n   C. Real-world examples of Java in secure applications\n      1. Banking
systems\n      2. Government and defense\n\nIV. Versatility and Diverse Applications\n   A. Enterprise
applications\n      1. Java EE and its role in enterprise architecture\n      2. Examples of large-scale
enterprise systems\n   B. Mobile applications\n      1. Java\'s role in Android development\n      2. Case
study of popular Android applications developed in Java\n   C. Web development\n      1. Java-based
frameworks like Spring\n      2. Use in scalable web services and applications\n   D. Emerging
technologies\n      1. Adoption in Internet of Things (IoT)\n      2. Role in cloud computing and big
data\n\nV. Extensive Ecosystem and Community Support\n   A. Wide range of libraries and
frameworks\n      1. Explanation of open-source contributions\n      2. Popular Java frameworks and
tools: Spring, Hibernate, Maven\n   B. Strong and active community\n      1. Java Community Process
(JCP)\n      2. Support networks and resources, including forums and educational sites\n   C.
Contribution to innovations and industry standards\n\nVI. Continuous Adaptability and Evolution\n
A. Regular updates and enhancements\n      1. Overview of Java\'s major version releases\n      2.
Introduction of new features: lambda expressions, streams, modules\n   B. Maintaining relevance in
modern software development\n   C. Future prospects of Java in technology trends\n      1. Anticipated
developments in AI and machine learning\n      2. Role in future-proof software solutions\n\nVII.
Criticisms and Challenges\n   A. Performance considerations against lower-level languages\n   B.
Complexity in comparison to newer languages\n   C. Discussion on overcoming these
challenges\n\nVIII. Conclusion\n   A. Recap of Java’s core strengths and contributions to technology\n
B. Personal reflection on Java’s impact on the author or industry professionals\n   C. Closing thoughts
on the enduring greatness of Java in a rapidly evolving technological landscape\n\nIX. References\n
A. Citing books, articles, and online resources on Java\n   B. Inclusion of quotes and insights from
industry experts\n   C. Reference to Java documentation and technical specifications\n\nThis outline
provides a comprehensive structure for exploring Java\'s greatness, considering its historical context,
technical advantages, and influence on diverse technological sectors.', 'story': '**The Greatness of Java:
The Time-Tested Programming Language**\n\nJava, originating in the mid-1990s under the auspices
of Sun Microsystems, has transformed from a fledgling language to a cornerstone of modern software
development. As tech industries burgeon with numerous programming languages, Java’s enduring
relevance stems from its platform independence, robust ecosystem, versatility, and continuous
adaptability.\n\nJava’s mantra, “Write Once, Run Anywhere” (WORA), is enabled by the Java Virtual
Machine (JVM), which allows portability across diverse hardware and operating systems. This ability
has empowered countless applications across sectors, demonstrating Java’s unparalleled flexibility and
resilience in evolving tech landscapes.\n\nRobustness and security are hallmarks of Java’s design. Its
automatic memory management and stringent type checking minimize common programming errors,
while features like bytecode verification and comprehensive security APIs ensure secure application
execution. Java’s reliability makes it a preferred choice for critical systems in banking, government,
and defense, where security cannot be compromised.\n\nJava’s versatility is evident in its widespread
applications. From powering large-scale enterprise systems through Java EE to driving innovation in
```

mobile development, particularly in Android, Java’s impact is ubiquitous. In web development, frameworks like Spring allow the creation of scalable services, while its emergence in IoT, cloud computing, and big data underscores its adaptability to new technologies.\n\nThe Java community bolsters its ecosystem, with extensive libraries, tools, and active forums that drive constant innovation and maintain its industry standard status. Regular updates bring features like lambda expressions and modules, ensuring Java remains at the forefront of software development, with anticipated roles in AI and machine learning.\n\nDespite challenges like performance gaps with lower-level languages and complexities compared to newer syntaxes, Java’s adaptability ensures it overcomes such hurdles. Its sustained evolution assures its place in future-proof software solutions.\n\nIn conclusion, Java’s core strengths, its profound industry impact, and its ability to adapt to ever-changing technological demands highlight its enduring greatness. For professionals and enthusiasts alike, Java’s legacy is not just a testament to its past achievements but a beacon guiding future technological advancements in an incessantly evolving landscape.'}