Task no. 2: Compare the Two Computing Models: Mainframe vs. Cloud

Mainframe Computing:

- **Centralized Processing**: Mainframes are powerful machines used for large-scale computing tasks, centralized in a single location.
- **High Reliability and Security**: Mainframes are known for their robustness, reliability, and high security, often used in banking, insurance or airports
- **Resource Allocation**: Mainframes allow for efficient resource allocation and handle a vast number of transactions simultaneously
- **Expensive Infrastructure**: Setting up and maintaining mainframe infrastructure is costly
- **Limited Flexibility**: Scaling up involves investment in hardware and often physical space

Cloud Computing:

- **Distributed Processing**: Cloud computing combines a distributed network of servers hosted on the internet to manage, stor and process data
- **Global Accessibility**: Services and data can be accessed from anywhere, facilitating remote work and global operations.
- **Cost-Effective**: Pay-as-you-go models reduce upfront costs and ongoing operational expenses.
- **Scalability and Flexibility**: Cloud services offer on-demand resource allocation, making it easy to scale up or down based on demand

Why did it changed over time so dramatically?

- **Cost Efficiency**: Cloud computing reduces operational costs, making it accessible for businesses of all sizes
- **Flexibility and Scalability**: The ability to quickly scale resources to match workload demands is a significant advantage over the fixed capacity of mainframes.
- **Technological Advancements**: Improvements in internet speed, security protocols, and virtualization technologies have made cloud computing more viable
- **Business Agility**: Cloud computing supports agile methodologies, enabling faster development cycles and innovation.

Task no. 2: Summary of Cloud Architecture Advantages

1. **Almost Zero Upfront Infrastructure Investment**:
    - Cloud services eliminate the need for initial heavy investments in hardware and infrastructure by offering pay-as-you-go models.
    - My code can be deployed on cloud platforms, removing the need for significant upfront investments in infrastructure.

2. **Just-in-Time Infrastructure**:
    - Cloud systems can quickly adjust resources up or down based on demand, avoiding the problem of having too much or too little capacity.

    > The notification system in my code can dynamically scale resources in the cloud, ensuring it handles varying loads efficiently.

3. **More Efficient Resource Utilization**:
    - Applications can dynamically use and release resources in the cloud, ensuring resources are not wasted and are used efficiently

    > My system can dynamically allocate and release resources, improving efficiency and reducing waste

4. **Usage-Based Costing**:
    - Customers pay only for the actual resources they use in the cloud, making it a cost-effective solution.

    > Deploying my application in the cloud allows you to pay only for the resources used during website monitoring and notifications, making it cost-effective.

5. **Potential for Shrinking Processing Time**:
    - Cloud services can run tasks in parallel, significantly speeding up processing times for large computations.

    > My system can utilize cloud-based parallel processing to check multiple websites simultaneously, reducing the time required for updates.

6. **Scalable Ingredients and Loosely Coupled Systems**:
    - Cloud applications are built with components that can scale independently and are loosely connected, enhancing reliability and scalability.

    > The Observer and Strategy patterns in my code ensure components are loosely coupled and can scale independently, aligning well with cloud architecture principles

7. **Resilience to Reboot and Re-Launch**:
    - Cloud systems are designed to recover automatically from hardware failures with features like backup and restore, ensuring continuous operation

    > Cloud deployment can provide automated recovery and backup features for my system, ensuring resilience against failures.

8. **Efficient Parallel Processing**:
    - Cloud computing allows for efficient parallel processing using multiple processors and nodes, simplifying complex computational tasks.

    > My code can take advantage of cloud services that support multi-threading and distributed processing, enhancing performance for complex tasks.

Task no. 3: Components and Cloud Services

1. **User Management**:
    - **Function**: Handles user authentication and management with built-in security features.
2. **Website Monitoring**:
    - **Function**: Executes serverless monitoring scripts for websites.
3. **Data Storage**:
    - **Function**: Stores user subscriptions, website data, and other relevant information.
4. **Notification Service**:
    - **Function**: Sends notifications via email, SMS, and push notifications.
5. **Application Logic**:
    - **Function**: Hosts the core application logic and web interfaces.
6. **Monitoring and Logging**:
    - **Function**: Provides comprehensive monitoring, logging, and alerting capabilities.

Task no. 4: AI

Example:

"Write a Python function that checks if a number is a prime number."

```
def is_prime(n):
    if n == 1:
        return True
    for i in range(2, n):
        if n % i == 0:
            return False
    return True
```

1. **Logical Error:**
   The function returns True for n = 1, but 1 is not a prime number. By definition, prime numbers are greater than 1 and divisible only by 1 and themselves.
2. **Inefficient Loop:**
   The loop runs from 2 to n-1. For optimization, we should only check up to √n (square root of n) for factors.

Correct Solution:

```
import math
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True
```