Question Answering Assignment

1. How can we handle security issues concerning connection strings written as hardcode ? Name 3 approaches to do this.

To handle the security issues related to hardcoded connection strings in .NET, here are three common approaches:

1. **Use Configuration Files with Encryption**
   Store connection strings in configuration files such as appsettings.json (for .NET Core) or web.config (for .NET Framework), but never hardcode them in code. For production, encrypt the connection strings section in the configuration file using built-in .NET features like Protected Configuration to prevent plain-text exposure.

2. **Use Secret Management Tools**
   In development and production, use secret management solutions such as ASP.NET Core User Secrets (for development), Azure Key Vault, or other secure vault services. These allow storing connection strings securely outside of the code and configuration files, retrieving them safely at runtime.

3. **Environment Variables**
   Store connection strings in environment variables on the hosting server or container environment. This avoids embedding sensitive data in code or files and does not rely on the application's config files directly. The app reads the connection string from environment variables at runtime.

These approaches avoid exposing sensitive database credentials and connection details directly in code, reducing the risk of unauthorized access or accidental leaks due to source control commits or decompiled binaries.