1. Why certain approaches were used, why others were not selected

Redis –

1. Redis is an open source BSD licensed advanced key-value cache store and persistent storage, which also referred to as data structure server as keys can contain a string, hashes, lists, sets, bitmap, sorted lists, and hyperloglogs
2. Fastest performance among other cache storage techniques. As it applies indexes on columns.
3. Support for SignalR, which enables the live processing and transformation of contents. This feature will be in high demand in the future so that it will satisfy future demands.
4. Data persist until the business specified time, in other caches data will vanish after some time.
5. Other in-memory caching techniques follows limited data structure, hence the performance will impact.
6. Redis allows storing key and value pairs as large as 512 MB.
7. Redis supports multiple programming languages, and capable to communicate regardless of which language does the last changes.
8. Redis offers data replication.
9. Supports big data platform.
10. Any design patterns used

Strategy Design Pattern-

1. Here project task belong to two types of project types, first is agile and other is waterfall project’s task.
2. So here we used two Strategy first agile strategy and other is a waterfall or normal project strategy.
3. Each type of projects can have different terms like agile has burned hours, story points whereas waterfall has estimated time, the severity of the task.
4. The implemented pattern capable to handle present changes and future changes.
5. The new type of project can also be created by doing changes at three places in the scope of C#.

Observer Design Pattern –

1. This is like triggers in SQL.
2. Notification sent after the execution of specific action like insert, update, delete and logged into the file.
3. So the developer has to concentrate on the main functionality, not on the other activity like logging the information.
4. Maintainance of logs will becomes smooth using the observer design pattern.
5. Anything extra you would have done given more time

Handled UI smoothly using jquery, please refer the following path,

~/ToDoList/ToDoContent/JS/AllTasks.js

1. Anything else you feel we should know

Before executing the project install Redis locally.

From C# side both types of projects (agile and normal/waterfall) have implemented, but on UI end only Normal/Waterfall project has handled due to time constraint.

All changes like insert, update, delete, delete multiple redirected without page refresh. For more information go through following file,

~/ToDoList/ToDoContent/JS/AllTasks.js



















