TITLE

**Developing a Client-Server Repository Synchronizer Application Inspired by Dropbox Functionality.**

# A CAPSSTONE PROJECTREPORT

***Submitted to***

**SAVEETHA SCHOOL OF ENGINEERING**

***By***

# SRINIVAS REDDY (192210136)

**S.JYOTHI KOUSHIK (192211914)**

# S. VENKATA SAI ABHIRAM

# (192210067)

**Supervisor**

**BY DR. MARY VALENTINA**

**SIMATS ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICALSCIENCES, CHENNAI – 602 105**

### Abstract:

### A server stores files. Distributed clients access the server, to learn about changes made to the files on the server, and to push local changes of the files onto the server. A synchronization application is used to synchronize the clients and server, synchronizing metadata and selected files. Client/Server computing is new technology that yields solutions to many data management problems faced by modern organizations. The term Client/Server is used to describe a computing model for the development of computerized systems.

### This model is based on distribution of functions between two types of independent and autonomous processes: Server and Client. A Client is any process that requests specific services from the server process. A Server is a process that provides requested services for the Client. Client and Server

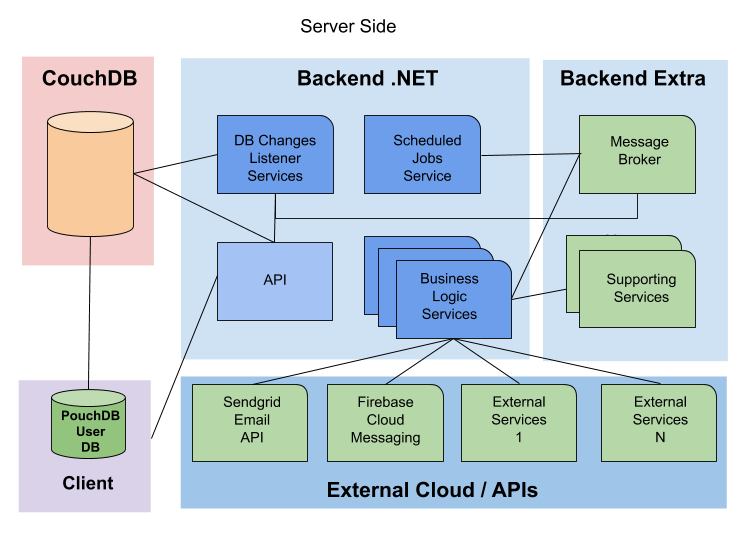
### Processes can reside in same computer or in different computers linked by a network. When Client and Server processes reside on two or more independent computers on a network, the Server can provide services for more than one Client. In addition, a client can request services from several servers on the network without regard to the location or the physical characteristics of the computer in which the

### Server process resides. The network ties the server and client together, providing the medium through which the clients and the server communicate. The Fig. 1.1 given below shows a basic Client/Server computing model.

# GANTT CHART

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 | Day 11 | Day 12 | Day 13 | Day 14 | Day 15 |
| Abstract and Introduction |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Literature survey |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Materials and Methods |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Results |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Discussion |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Repo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**FLOWCHART**



### INTRODUCTION

The modern digital landscape is marked by an ever-increasing need for seamless file synchronization and collaboration across multiple devices and locations. In response to this demand, the development of a Client-Server Repository Synchronizer Application inspired by Dropbox functionality emerges as a pivotal solution. Drawing inspiration from the renowned file hosting service, this application aims to empower users with the ability to effortlessly synchronize and share files between their devices and a centralized server. By incorporating features such as file monitoring, conflict resolution, and selective sync, this application endeavors to streamline file management while ensuring data integrity and accessibility. Through the convergence of robust client-side applications and scalable server-side infrastructure, this endeavor seeks to address the dynamic challenges of modern data synchronization with efficiency and reliability.

The exponential growth of digital data and the ubiquity of multiple devices in today's interconnected world underscore the necessity for a sophisticated solution to manage file synchronization and collaboration effectively. Inspired by the pioneering functionality of Dropbox, a Client-Server Repository Synchronizer Application emerges as a vital tool in addressing this imperative. This innovative application harnesses advanced technologies to facilitate seamless synchronization of files between a user's devices and a centralized server repository. Leveraging features such as real-time file monitoring, robust conflict resolution mechanisms, and customizable selective sync options, users can effortlessly maintain the consistency and accessibility of their data across platforms. With a user-centric approach and a focus on scalability and security, this solution endeavors to empower individuals and organizations alike to navigate the complexities of modern data management with confidence and ease.

OBJECTIVE

1. **Efficient File Synchronization**: Develop algorithms and mechanisms to ensure seamless and efficient synchronization of files between client devices and the server repository, minimizing transfer times and resource consumption.
2. **Secure Data Transmission and Storage**: Implement encryption protocols for secure data transmission over the network and ensure robust security measures for storing files on the server repository, safeguarding sensitive user data from unauthorized access or breaches.
3. **User-Friendly Interface**: Design an intuitive and user-friendly interface for

both the client-side application and the server-side administration dashboard,

enhancing user experience and simplifying configuration and management tasks.

**5.Scalability and Performance Optimization**: Architect the application to be highly scalable, capable of handling increasing numbers of users and large volumes of data efficiently. Implement performance optimization techniques to minimize latency and ensure responsive file synchronization.

## LITERATURE REVIEW

A comprehensive review of the literature surrounding Client-Server Repository Synchronizer Applications reveals a growing body of research and development aimed at addressing the complex challenges inherent in file synchronization and collaboration in distributed computing environments. Existing studies have explored various aspects of these applications, including synchronization algorithms, conflict resolution strategies, security protocols, and user interface design. One notable area of focus is the development of efficient synchronization algorithms that minimize data transfer overhead while ensuring the consistency and integrity of synchronized files across multiple devices. Research in conflict resolution has emphasized the importance of implementing robust mechanisms to detect and resolve conflicts that arise when multiple users attempt to modify the same file concurrently. Additionally, considerable attention has been devoted to enhancing the security of data transmission and storage, with encryption protocols and access control mechanisms being central to safeguarding sensitive user data. Furthermore, literature on user interface design has highlighted the significance of intuitive and user-friendly interfaces in facilitating seamless user interaction and adoption. Overall, the literature underscores the multifaceted nature of Client-Server Repository Synchronizer Applications and provides valuable insights into the design, implementation, and optimization of these systems to meet the evolving needs of modern users and organizations.