Backup and Recovery

Objective

To ensure data availability, integrity, and recoverability in the event of failures, a comprehensive backup and recovery strategy has been designed for the PostgreSQL database.

Backup Types & Schedule

Backup type	**Description**	**Frequency**	
Full Backup	A complete snapshot of the entire database	weekly	
Differential Backup	Captures changes made since the last full backup	Daily	
Transaction Log Backup	Captures all committed transactions since the last transaction log backup	Every 15-30 minutes	

Backup Implementation Tools

- **pg_dump**: For full logical backups (structured SQL dump)
- **pg_basebackup**: For physical full database backup (used with WAL archiving)
- **WAL Archiving**: PostgreSQL's Write-Ahead Logging allows continuous archiving of changes for PITR
- **cron jobs** or **pgBackRest / Barman** for automation and consistency

Failure Handling Strategy

Failure Scenario	**Recovery Action**
Accidental Data Deletion	Restore latest transaction log and roll forward from last full + differential backup

Server Crash	Reboot database using last physical backup + WAL recovery
Disk corruption / Loss	Restore latest full backup from remote / cloud server + WAL
Application Level Bugs	Roll back using PITR to a known good timestamp

3. Unauthorized Access Prevention

Security Measure	**Description**
Role based Access control	Assign roles with only necessary permissions to users
Password Authentication	Enforce strong password policies and hashing mechanisms
SSL/TLS Encryption	Encrypts all client server communication
Firewall and IP whitelisting	Allow access only from trusted IP addresses
Regular Monitoring and Auditing	Log access and changes, review logs of suspicious activity

Storage and Retention Policy

- Backups are stored in:
    - **Primary local server**
    - **Secondary remote server/cloud (e.g., AWS S3/Dropbox/Google Drive)**
- **Retention period**:
    - Full backups: 4 weeks
    - Differential: 1 week
    - Logs: 1 week minimum (configurable)

Security Measures

- Encrypt all backup files using **GPG or OpenSSL**
- Use **role-based access control** to restrict backup operations
- Enable **SSL connections** for data transfer between DB and backup storage
- Regular **integrity checks** (checksum validation)

Documentation and Monitoring

- Maintain a **backup log file** for every operation (timestamped)
- Setup **monitoring alerts** for failed backups or abnormal job durations
- Run **monthly recovery drills** to ensure backups are restorable

Conclusion

This backup and recovery strategy ensures that the chat application's PostgreSQL database remains secure, available, and recoverable at all times. With automated differential and transaction log backups, we ensure minimal data loss in case of failure and fast restoration to reduce downtime. In addition, by enforcing robust security measures such as role based access control, encrypted connections, and access monitoring we protect sensitive data from unauthorized access and breaches. Together, these measures support the system's reliability and security, crucial for user trust and operational success.