**Journal of Computer and Electrical Sciences (JCES)- Volume-1 Issue-4, April 2026**
**ISSN: 3049-2602**

# DATABASE BACKUP AND RESTORE MANAGER

**B. Raghavendra[1], L. Satish[2], G. Varun[3], A. V N S Aditya Saran[4], CH. Sri Hareeshwar Reddy[5], Dr. K. Sireesha[6]**

*[1,2,3,4,5]Student, Department of Computer Science & Engineering*
*Andhra Loyola Institute of Engineering and Technology, ITI road, Vijayawada, Andhra Pradesh, India*
*[6]Associate Professor, Department of Computer Science & Engineering*
*Andhra Loyola Institute of Engineering and Technology, ITI road, Vijayawada, Andhra Pradesh, India*
*Email id: raghavendrabekkanti4@gmail.com, lamsatish5h7@gmail.com, gurrala.varun11@gmail.com, adityasaran734@gmail.com, sriharicherukula@gmail.com,*

***Abstract:*** *In many organizations, database backups are still performed manually using command-line tools or scripts. These methods require technical expertise, are time-consuming, and are highly prone to human errors such as incorrect commands, missed schedules, or improper storage of backup files. As databases continue to grow in size and play a critical role in business operations, unreliable or poorly managed backup practices can lead to serious data loss, system downtime, and operational disruptions. To address these challenges, this project presents a web-based Database Backup and Restore Manager developed using Spring Boot and Full Stack Java technologies. The application provides a user-friendly web interface that simplifies the process of managing database backups without requiring deep command-line knowledge. It supports multiple database management systems such as MySQL, PostgreSQL, and MongoDB making it suitable for diverse environments. The system offers essential features including automated and scheduled backups, compression of backup files to save storage space, support for both local and cloud-based storage, and detailed logging of backup activities for monitoring and auditing purposes. Additionally, the application enables reliable restore operations to recover databases when required. By centralizing and automating backup operations through a web platform, the proposed solution improves data safety, reduces manual effort, and enhances overall efficiency in database administration.*

## 1. INTRODUCTION

In the modern digital era, organizations and individuals rely heavily on databases to store and manage critical information. Databases are widely used in applications such as banking systems, healthcare systems, educational platforms, and e-commerce websites. As the amount of digital data continues to grow rapidly, ensuring the availability, reliability, and security of stored data has become an important requirement. One of the most effective methods to protect valuable data is through database backup and recovery mechanisms. Database backup is the process of creating copies of database data so that the information can be restored in case of data loss or system failure. Data loss can occur due to hardware failures, software bugs, accidental deletion, system crashes, or cyber-attacks. Without proper backup mechanisms, organizations may lose critical information that can lead to operational disruptions and financial losses. Traditional backup methods often rely on command-line utilities and manual processes. Although these tools are useful, they can be difficult to manage, especially when handling multiple database systems. Administrators must execute commands manually, maintain backup files, and track backup history separately. These limitations make it challenging to manage backups efficiently in environments where several databases are used simultaneously. With the advancement of web technologies, modern solutions

are shifting toward web-based backup management systems that provide centralized control and user-friendly interfaces. Such systems allow administrators to configure database connections, schedule backups, monitor backup activities, and restore data through graphical dashboards instead of complex command-line operations. The proposed system in this project introduces a web-based database backup and recovery platform that supports multiple database systems such as MySQL, PostgreSQL, and MongoDB. The system allows administrators to perform full and incremental backups, restore databases when required, and monitor backup operations through an intuitive web interface. The backend of the system is implemented using Spring Boot, which provides robust REST APIs for managing backup and restore operations. The frontend interface is developed using Vue.js, enabling administrators to interact with the system through a modern and responsive user interface.The primary objective of this project is to design and develop a reliable and easy-to-use database backup management system that simplifies the process of protecting and restoring data while supporting multiple database platforms.

## 2. LITERATURE SURVEY

Database backup and restore manager is an essential component of modern database management systems. Organizations rely heavily on databases to store critical information, and therefore reliable backup systems are required to prevent data loss caused by hardware failures, system crashes, or accidental deletion. Several researchers have studied different techniques to improve database backup performance, reliability, and recovery efficiency. One of the commonly used approaches in database systems is physical and logical backup mechanisms. Physical backup involves copying the raw database files directly from the storage system, while logical backup extracts the data from the database and stores it as a sequence of SQL statements that can recreate the database structure and content. Both techniques support full backups that store the entire database and incremental backups that store only the changes made since the previous backup. However, these methods often require additional input/output operations, which can increase backup time and recovery overhead. Several tools have been developed to support these backup strategies. For example, utilities such as Xtrabackup perform physical backups by copying database files, while tools like MySQLdump generate logical backups by scanning database tables and creating SQL queries that reproduce the data during recovery. Although these tools are widely used, they may require significant time and system resources when dealing with large databases. To address these limitations, researchers have explored the use of Write-Ahead Logging (WAL) for backup and recovery operations. WAL is a logging technique used in many database systems to maintain data consistency and durability. In this approach, all modifications to the database are first recorded in a log file before being written to the actual data files. Because WAL contains all database changes, it can be utilized to reconstruct the database state during recovery operations. The research paper titled “An Efficient Database Backup and Recovery Scheme using Write-Ahead Logging” proposes an approach that utilizes existing WAL logs to perform backup operations without additional data scanning or copying. This technique reduces backup time and improves recovery speed by applying log data directly during restoration. The system also optimizes the recovery process by dividing recovery operations into smaller ranges and 5 processing them using multiple threads to improve performance. In addition to WAL-based techniques, other studies have explored snapshot-based backup mechanisms. File systems such as BTRFS and Ext3cow implement snapshot features using copy-on-write strategies to maintain multiple versions of stored data. These techniques allow administrators to restore data to a specific point in time. However, snapshot-based systems may introduce additional write operations and may depend on specific file system implementations. Recent developments in backup management systems

focus on automation and improved usability. Modern systems integrate automated backup scheduling, cloud storage integration, and monitoring dashboards to simplify database administration tasks. Web-based backup management tools developed using frameworks such as Spring Boot for backend services and Vue.js for frontend interfaces allow administrators to manage backup operations through graphical dashboards instead of command-line utilities. The proposed system builds upon these research efforts by implementing a web-based database backup and recovery platform that supports multiple database systems and provides automated backup scheduling, compression of backup files, logging of backup activities, and restore operations. By combining modern web technologies with efficient backup strategies, the system aims to provide a reliable and user-friendly solution for database backup management.

# 3. PROPOSED SYSTEM

The proposed system introduces a web-based database backup management platform developed using Spring Boot for backend services and Vue.js for frontend interface. Unlike traditional command-line tools, the proposed system provides a graphical user interface that allows administrators to easily manage backup operations.

The main features of the proposed system include:

1.Support for multiple database management systems
2.Automated backup scheduling
3.Full and incremental backup operations
4.Compression of backup files
5.Backup storage in local or cloud environments
6.Backup activity logging
7.Database restore functionality

The system improves usability by providing a centralized dashboard where administrators can monitor backup activities, view logs, and perform restore operations. By integrating modern web technologies and database management techniques, the proposed system offers a more efficient and user-friendly solution for database backup and recovery management.

**Comparison of Existing Works with Proposed System**

| Ref No | Work / System | Automated Backup | Web UI / GUI | Multi-DB Support | Scheduling | Compression / Storage Optimization | Cloud Storage Support | Logging / Monitoring | Restore Support |
|---|---|---|---|---|---|---|---|---|---|
| [1] | Bhattacharya et al. – Coordinating Backup & Recovery | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| [2] | Amvrosiadis & Bhadkamkar – Enterprise Data Protection Trends | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| [3] | Amvrosiadis & Bhadkamkar – Backup Failure Analysis | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| [4] | Kaiser et al. – Sorted Deduplication Backup System | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ |
| [5] | Schwartz et al. – MySQL Backup Techniques | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| [6] | Son et al. – SSD Assisted Backup & Recovery | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| | **Proposed System** Web-based Database Backup & Restore Manager | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Fig 3.1: Existing vs Proposed

# 4. METHODOLOGY

The proposed system is implemented using multiple modules that work together to perform database backup and recovery operations. The modules in the system include: Administrator Module
Backup Engine Module
Restore Engine Module
Scheduling Module
Logging and Monitoring Module
MODULES DESCRIPTION

**Administrator Module:** The Administrator module provides access to the system through a web interface. The administrator can configure database connections for MySQL, PostgreSQL, and MongoDB. The administrator can initiate backup operations, restore data from existing backups, schedule automated backups, and monitor system activity.

The administrator dashboard displays statistics such as total backups performed, failed backups, successful restore operations, and system logs.

**Backup Engine Module:** The Backup Engine module is responsible for creating database backups. It supports both full backup and incremental backup methods. In full backup, the entire database schema and data are exported. In incremental backup, only the records that have been updated after the last backup are exported. The backup engine interacts with different database drivers depending on the database type: 22 JDBC for MySQL and PostgreSQL MongoDB Java Driver for MongoDB the generated backup files are stored in the local file system, and metadata about the backup operation is stored in the metadata database.

**Restore Engine Module:** The Restore Engine module is responsible for restoring database data from backup files. When a restore request is initiated, the system identifies the appropriate backup group. The system first restores the latest full backup and then applies any incremental backups that were created after the full backup. This process ensures that the database is restored to the most recent consistent state.

**Scheduling Module:** The Scheduling Module allows administrators to automate backup operations. The system supports scheduled backups at different intervals such as hourly, daily, or weekly. The scheduling service runs in the backend and automatically triggers backup operations based on the configured schedule.

**Logging and Monitoring Module:** The Logging and Monitoring Module records all activities related to backup and restore operations. The system logs both successful and failed operations. The logs are stored in the metadata database and can be viewed through the administrator dashboard. This helps administrators monitor system performance and troubleshoot errors.

# 5. PROPOSED SYSTEM RESULTS

The proposed Database Backup and Restore Manager was successfully implemented and tested under multiple operational scenarios. The system effectively performs backup, restore, scheduling, and monitoring operations in real time.

- The system accurately performs **database backup operations** for multiple databases. Backup files are generated successfully and stored with proper timestamps and metadata.

- The **restore functionality** works reliably by retrieving previously stored backups and reconstructing the database without data loss.
- The **dashboard module** displays real-time information such as total backups, successful operations, failed attempts, and recent activities, providing an efficient monitoring interface.
- The system successfully detects and reports **errors during backup and restore operations**, enabling quick identification and resolution of issues.
- The **scheduler module** executes automated backups at predefined intervals without manual intervention, ensuring continuous data protection.
- The **log management system** maintains detailed records of all operations, including timestamps, status, and activity history for auditing purposes.
- The user interface is **responsive and user-friendly**, allowing smooth navigation between modules such as dashboard, backup, restore, and logs.

**Dashboard Interface:** The dashboard provides a centralized view of system performance, including backup statistics, system alerts, and recent activity logs.

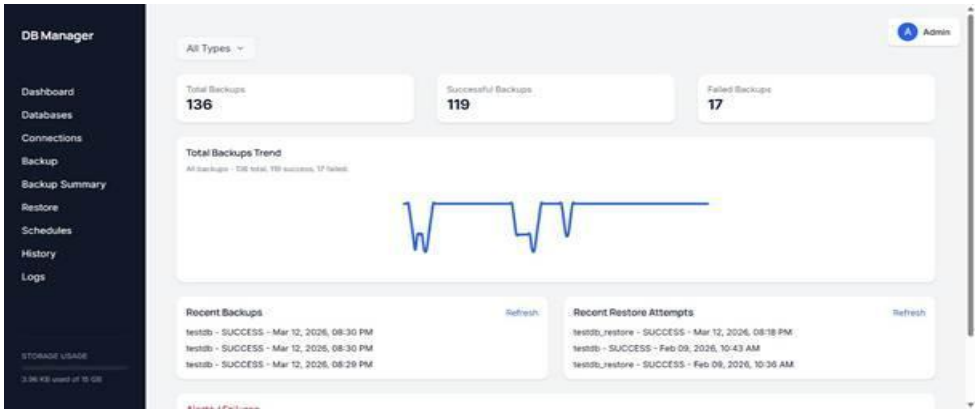


Fig 5.1: Dashboard

**Backup Operation:** The system successfully initiates and completes backup operations. The results show that backup files are created, stored, and logged correctly with accurate status updates.

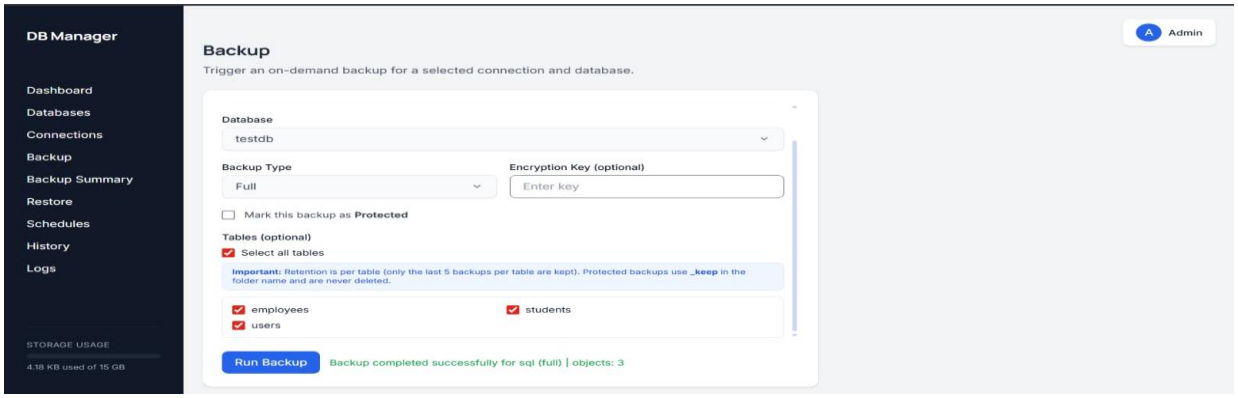


Fig 5.2: Backup Page

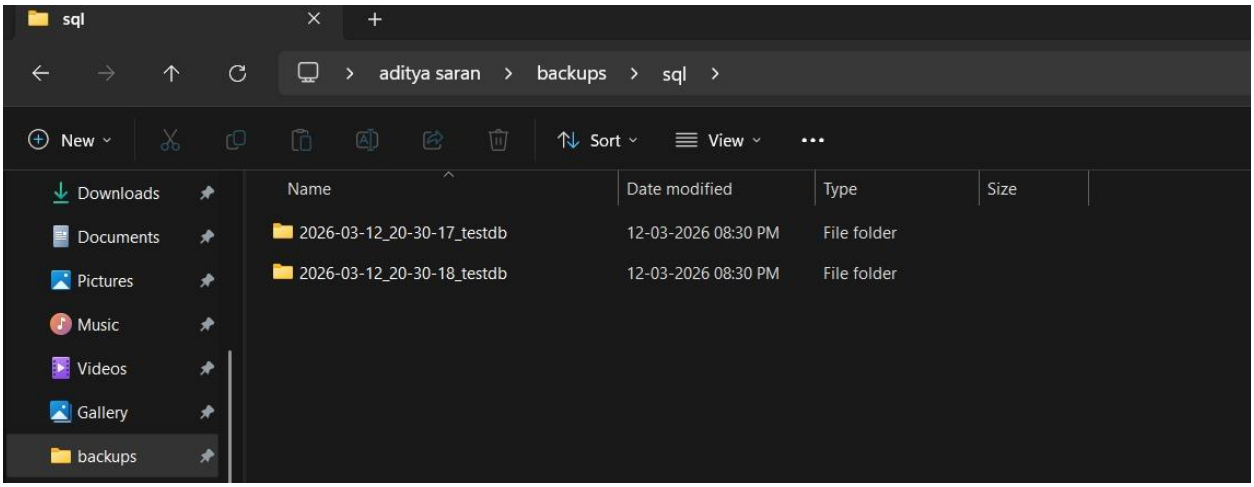


Fig 5.3: Backup Groups

**Restore Operation:** The restore module accurately retrieves backup data and restores database states. The system ensures data integrity and confirms successful execution after each restore operation.

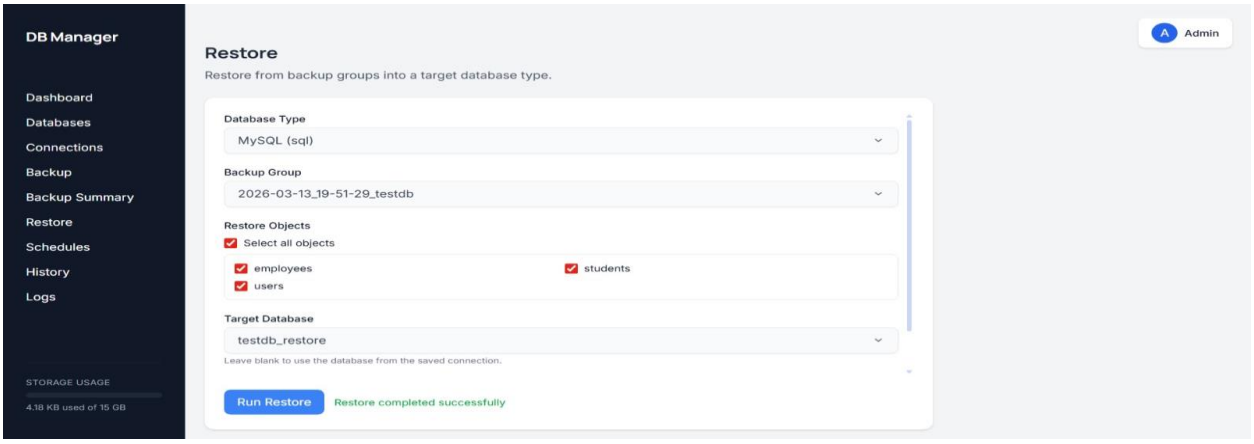


Fig 5.4: Restore page

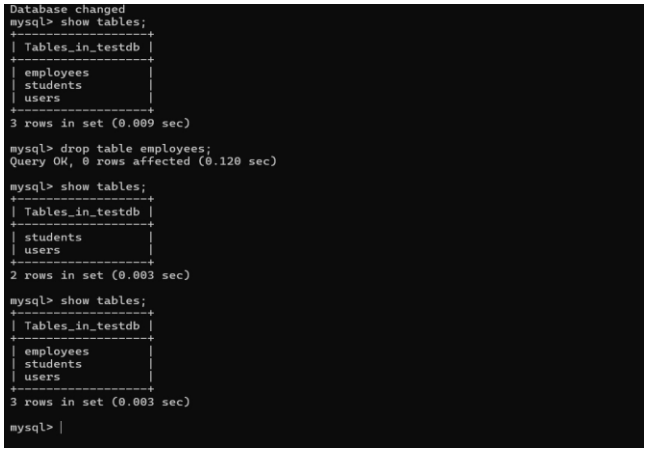


Fig 5.5: Restore Result

**Schedule Operation:** Scheduled backup tasks are executed at predefined times. The system performs automated operations consistently without delays or failures.

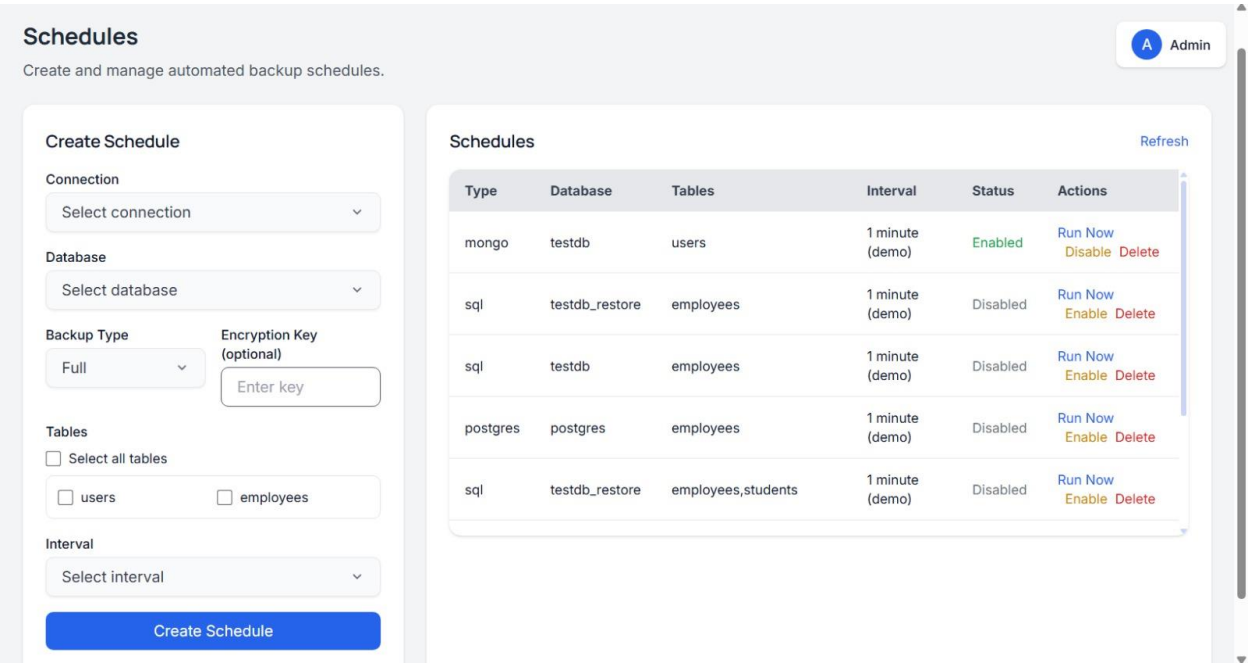


Fig 5.6: Schedule Page

**Logs and System Monitoring:** The logs section displays detailed records of all activities, including successful and failed operations. This improves traceability and system transparency.

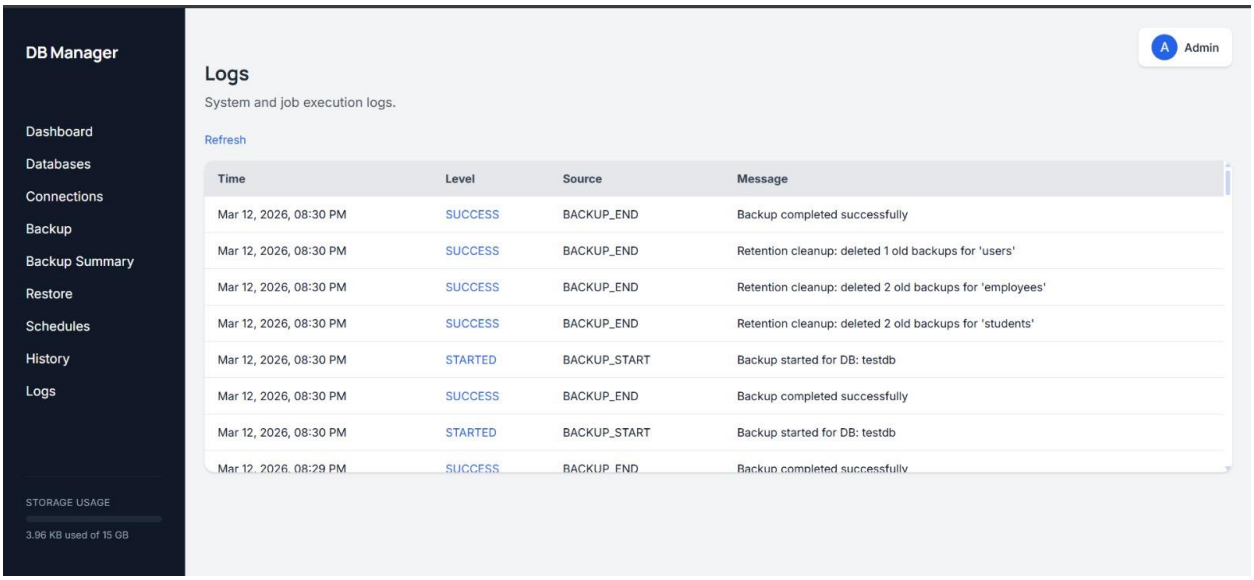


Fig 5.7: Logs

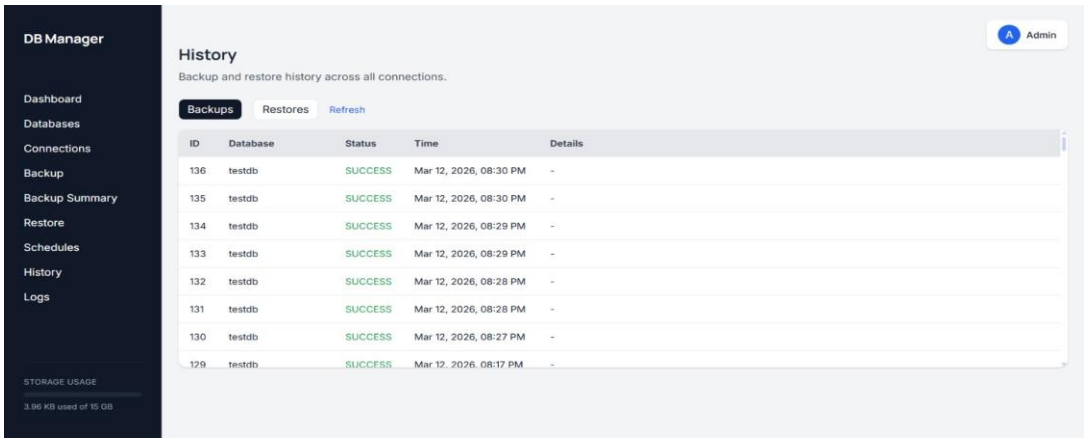


Fig 5.8: History

# 6. CONCLUSION

The Web-based Database Backup and Restore Manager was developed to simplify and automate the process of database backup and recovery. Traditional backup methods often depend on manual command-line tools, which require technical expertise and can lead to errors, missed backups, or data loss. The proposed system provides a web-based platform built using Spring Boot and full stack technologies, allowing users to manage backups through a simple and user-friendly interface. It supports multiple database systems such as MySQL, PostgreSQL, and MongoDB making it flexible for different environments. The system includes features such as automated backup scheduling, backup file compression, local and cloud storage support, and logging of backup activities. It also provides reliable restore functionality to recover databases when required. Overall, the system improves data security, efficiency, and reliability by automating and centralizing database backup management.

## REFERENCES

*1. H. Kim, H. Y. Yeom, and Y. Son, "An Efficient Database Backup and Recovery Scheme using Write-Ahead Logging," in Proc. 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), Virtual Event, Oct. 2020, pp. 405–413, IEEE.*

*2. Son, Y., Choi, J., Jeon, J., Min, C., Kim, S., Yeom, H. Y., and Han, H., "SSD-assisted backup and recovery for database systems," In 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 285–296, San Diego, CA, USA (2017)*

*3. Amvrosiadis, G., and Bhadkamkar, M., "Identifying trends in enterprise data protection systems," In 2015 USENIX Annual Technical Conference (USENIX ATC 15), pp. 151–164, Santa Clara, CA, USA (2015).*

*4. Amvrosiadis, G., and Bhadkamkar, M., "Getting back up: Understanding how enterprise data backups fail," In 2016 USENIX Annual Technical Conference (USENIX ATC 16), pp. 479–492, Denver, CO, USA (2016).*

*5. Kaiser, J., Süß, T., Nagel, L., and Brinkmann, A., "Sorted deduplication: How to process thousands of backup streams," In 2016 32nd Symposium on Mass Storage Systems and Technologies (MSST), pp. 1–14, Santa Clara, CA, USA (2016).*

*6. Rodeh, O., Bacik, J., and Mason, C., "Btrfs: The Linux B-tree file system," ACM Transactions on Storage (TOS), 9(3), pp. 1–32 (2013).*

*7. Peterson, Z., and Burns, R., "Ext3cow: A time-shifting file system for regulatory compliance," ACM Transactions on Storage (TOS), 1(2), pp. 190–212 (2005).*