Campus Insider

COS 457 / COS 557

Advanced Database Feature Documentation

Database Security & Authentication - Nathaniel

Where it's implemented:

- **Database**: MySQL; a dedicated low-privilege user account was created for the Campus Insider web application
- **Backend**: All SQL queries in the Flask backend use parameter binding (prepared statements) to prevent SQL injection

Effect:

- **Least-privilege enforcement**: The app can only access the data it needs; no administrative or destructive operations are possible
- **SQL injection projection**: User input is treated as data, not executable SQL, reducing the risk of malicious database manipulation

Backup and recovery - Benjamin

What is implemented:

- **Backup bash file (Mac OS):** the backup file prompts sqlDump for the database image
- **Backup directory:** once the backup file is executed and the correct password is input, then a backup image of the database is stored in the desired backup directory

- **Restore bash file (Mac OS):** the restore file takes a database and the image to replace it with. It then initialises the tables and inputs the new (old) data

Effect:

- **Backup and recovery usage:** the backup database file can be used for future recovery needs or to revert the database to a previous image in emergency situations.
- **Simple process:** the process is very lightweight due to using short bash files to get the backup image and restore it, however, the size of the backup file can be large depending on the size of the database schema and the data in the tables.