ijName – Aman Agrawal

PRN – 2020BTECS00074

Software Engineering Tools Lab Assignment No-1

(Module 1- Introduction to FOSS) **Due date-29/01/2023**

1. Differentiate in between free software, Open source software and proprietary software with respect to its properties.

|  |  |  |
| --- | --- | --- |
| S.No. | OPEN-SOURCE SOFTWARE | PROPRIETARY SOFTWARE |
| 01. | Open-source software is computer software whose source code is available openly on the internet and programmers can modify it to add new features and capabilities without any cost. | Proprietary software is computer software where the source codes are publicly not available only the company which has created can modify it. |
| 02. | Here the software is developed and tested through open collaboration. | Here the software is developed and tested by the individual or organization by which it is owned not by the public. |
| 03. | In open-source software the source code is public. | In proprietary software, the source code is protected. |
| 04. | Open-source software can be installed on any computer. | Proprietary software can not be installed into any computer without a valid license. |
| 05. | Users do not need to have any authenticated license to use this software. | Users need to have a valid and authenticated license to use this software. |
| 06. | Open-source software is managed by an open-source community of developers. | Proprietary software is managed by a closed team of individuals or groups that developed it. |
| 07. | It is more flexible and provides more freedom which encourages innovation. | It is not much flexible so there is a very limited innovation scope with the restrictions. |
| 08. | Users can get open software free of charge. | Users must have to pay to get the proprietary software. |
| 09. | In open-source software faster fixes of bugs and better security are availed due to the community. | In proprietary software, the vendor is completely responsible for fixing malfunctions. |
| 10. | Limited Intellectual Property Protections | Full Intellectual Property Protections |
| 11. | Usually Developed and Maintained by non-profit organizations. | Usually Developed and Maintained by for-profit entities. |
| 12. | Examples are Android, Linux, Firefox, Open Office, GIMP, VLC Media player, etc. | Examples are Windows, macOS, Internet Explorer, Google Earth, Microsoft Office, Adobe Flash Player, Skype, etc. |

1. Enlist some examples along with its purpose and properties (at least 10) of FOSS and proprietary software with respect to database.

* FOSS Database System:

1. MYSQL:

Purpose – it is a database management system and uses SQL (sequential query language) for operations. It is a solid, quick, and dependable jack-of-all-trades.

Properties:

* + It is open source.
  + It is free.
  + Highly Scalable
  + Elastic Replication
  + Secure

1. MariaDB:

Purpose - MariaDB is a [backward compatible](https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/), drop-in replacement of the MySQL® Database Server. It includes all major open source storage engines. The MariaDB database is used for various purposes such as data warehousing, e-commerce, enterprise-level features, and logging applications.

Properties:

* + Modern SQL and JSON support
  + Oracle Database compatibility, continuous availability and comprehensive security.
  + MariaDB's versatility stems from its pluggable storage engine architecture, with different storage engines optimized for different workloads.

1. PostgreSQL:

Purpose - PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

Properties:

* **Data Types**
  + Primitives: Integer, Numeric, String, Boolean
  + Structured: Date/Time, Array, Range / Multirange, UUID
  + Document: JSON/JSONB, XML, Key-value (Hstore)
  + Geometry: Point, Line, Circle, Polygon
  + Customizations: Composite, Custom Types
* **Data Integrity**
  + UNIQUE, NOT NULL
  + Primary Keys
  + Foreign Keys
  + Exclusion Constraints
  + Explicit Locks, Advisory Locks
* **Concurrency, Performance**
  + Indexing: B-tree, Multicolumn, Expressions, Partial
  + Advanced Indexing: GiST, SP-Gist, KNN Gist, GIN, BRIN, Covering indexes, Bloom filters
  + Sophisticated query planner / optimizer, index-only scans, multicolumn statistics
  + Transactions, Nested Transactions (via savepoints)
  + Multi-Version concurrency Control (MVCC)
  + Parallelization of read queries and building B-tree indexes
  + Table partitioning
  + All transaction isolation levels defined in the SQL standard, including Serializable
  + Just-in-time (JIT) compilation of expressions
* **Reliability, Disaster Recovery**
  + Write-ahead Logging (WAL)
  + Replication: Asynchronous, Synchronous, Logical
  + Point-in-time-recovery (PITR), active standbys
  + Tablespaces
* **Security**
  + Authentication: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificate, and more
  + Robust access-control system
  + Column and row-level security
  + Multi-factor authentication with certificates and an additional method
* **Extensibility**
  + Stored functions and procedures
  + Procedural Languages: PL/pgSQL, Perl, Python, and Tcl. There are other languages available through extensions, e.g. Java, JavaScript (V8), R, Lua, and Rust
  + SQL/JSON path expressions
  + Foreign data wrappers: connect to other databases or streams with a standard SQL interface
  + Customizable storage interface for tables
  + Many extensions that provide additional functionality, including PostGIS
* **Internationalisation, Text Search**
  + Support for international character sets, e.g. through ICU collations
  + Case-insensitive and accent-insensitive collations
  + Full-text search

1. Redis (Remote Dictionary Server)

Purpose: Redis is a data structure server. At its core, Redis provides a collection of native data types that help you solve a wide variety of problems, from caching to queuing to event processing.

Properties:

* In-memory data store to ingest, process, and analyze real-time data with sub-millisecond latency.
* Redis delivers sub-millisecond response times, enabling millions of requests per second for real-time applications in industries like gaming, ad-tech, financial services, healthcare, and IoT.
* Redis employs a primary-replica architecture and supports asynchronous replication where data can be replicated to multiple replica servers.

1. SQLite:

Purpose: SQLite is an embedded, server-less relational database management system. It is an in-memory open-source library with zero configuration and does not require any installation. Also, it is very convenient as it’s less than 500kb in size, which is significantly lesser than other database management systems.

Properties:

* SQLite is serverless as it doesn't need a different server process or system to operate.
* SQLite facilitates you to work on multiple databases on the same session simultaneously, thus making it flexible.
* SQLite is a cross-platform DBMS that can run on all platforms, including macOS, Windows, etc.
* SQLite doesn't require any configuration. It needs no setup or administration.

1. Neo4j:

Purpose: Neo4j is a graph database. A graph database, instead of having rows and columns has nodes edges and properties. It is more suitable for certain big data and analytics applications than row and column databases or free-form JSON document databases for many use cases. A graph database is used to represent relationships.

Properties:

* It is ACID (atomicity, consistency, isolation, and durability) compliant.
* It encourages versatility.
* Replicates information with quality and security.
* It works with Web applications for recovering chart information.
* It bolsters enquiry information sent out to JSON and XLS design.

1. OrientDB:

Purpose: OrientDB is an Open Source NoSQL Database Management System. NoSQL Database provides a mechanism for storing and retrieving NO-relation or NON-relational data that refers to data other than tabular data such as document data or graph data.

Properties:

* It is written in JAVA
* Zero Configuration Multi-master Architecture: This provides horizontal scalability and reliability.
* Multi-language support
* Multi-model support
* Zero delay start

1. CouchDB:

Purpose: CouchDB enables applications to store collected data locally on mobile devices and browsers, then synchronizes that data once it is back online.

Properties:

* An open source NoSQL document database that collects and stores data in JSON-based document formats
* Replication: One of CouchDB’s defining features is bi-directional replication, which enables synchronization of data across multiple servers and devices via bi-directional replication.
* Views: CouchDB uses views as the primary tool for running queries and creating reports from stored document files. Views allow you to filter documents to find information relevant to a particular database process.
* HTTP API: CouchDB uses a REST API to access the database from anywhere, with full CRUD (create, read, update, delete) operations flexibility.
* Built for offline: When you are scaling your database usability and accessibility, being able to build applications that work as well offline as they do online is essential.

1. FirebirdSQL:

Purpose: Firebird is a relational database offering many ANSI SQL standard features that runs on Linux, Windows, and a variety of UNIX platforms. Firebird offers excellent concurrency, high performance, and powerful language support for stored procedures and triggers.

Properties:

* It offers a small footprint, which is handy where space is premium.
* You can use FirebirdSQL as the database for a desktop app that needs to scale, such as LibreOffice.
* The database is cross-platform and has low needs in terms of hardware. This makes it a solid solution when you need to run large databases.

1. BigchainDB:

Purpose: BigchainDB is a blockchain database offering decentralization, immutability and native assets. BigchainDB allows for the deployment of large-scale applications in a variety of use cases and industries from intellectual property and identity to supply chain, and Internet-of-Things.

Properties:

* Scale (throughput, capacity, low latency)
* Queryability.
* Decentralized (no single entity owns or controls it)
* Immutable (tamper-resistance)
* Assets (you own the asset if you own the private key, aka blockchain-style permissioning).

1. Enlist some examples of free open source exam software for online assessment.

* FOSS for online assessment:

TCExam

VirtualX

Moodle

TAO

Kaldin

Papershala

Edbase

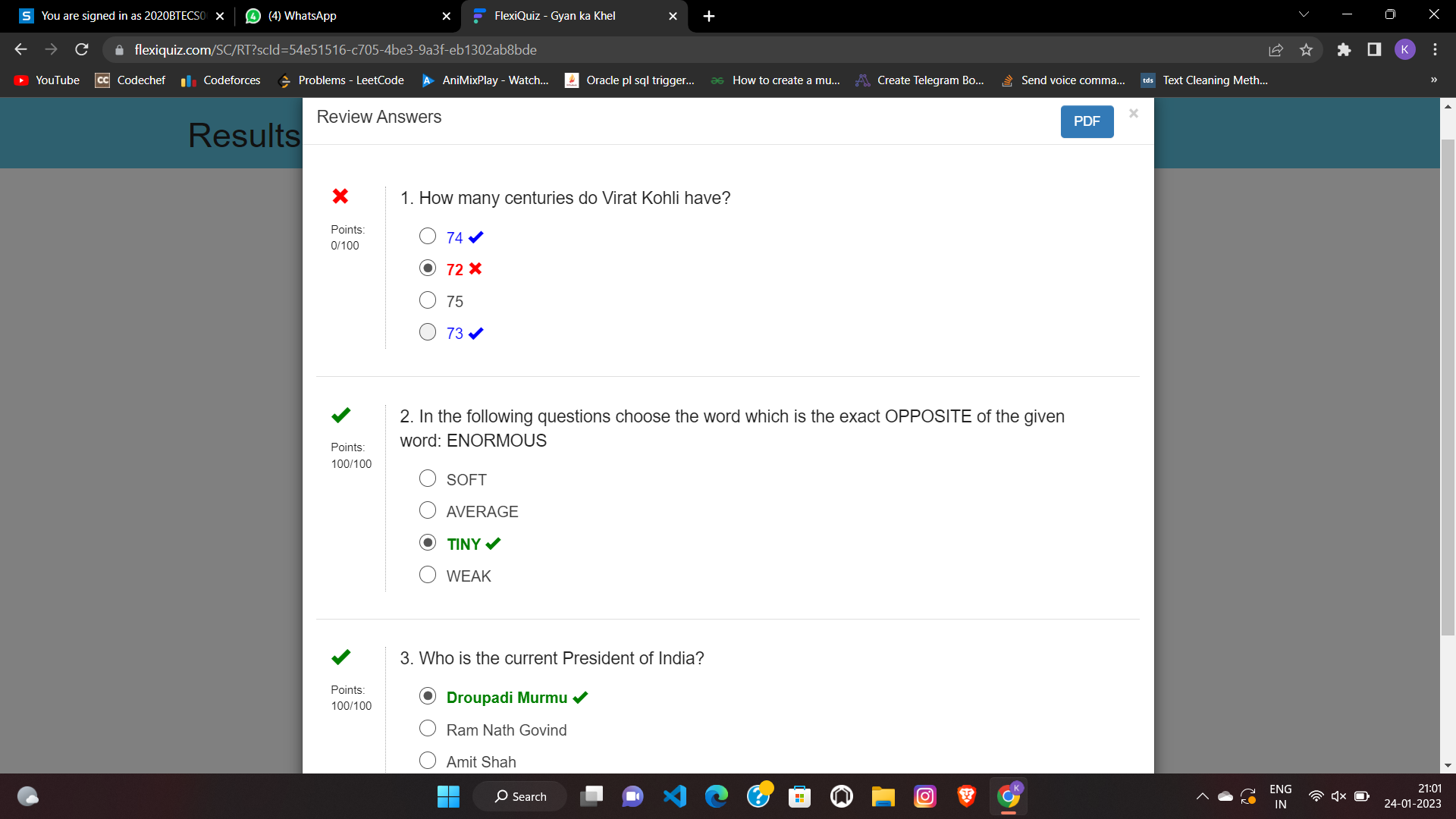
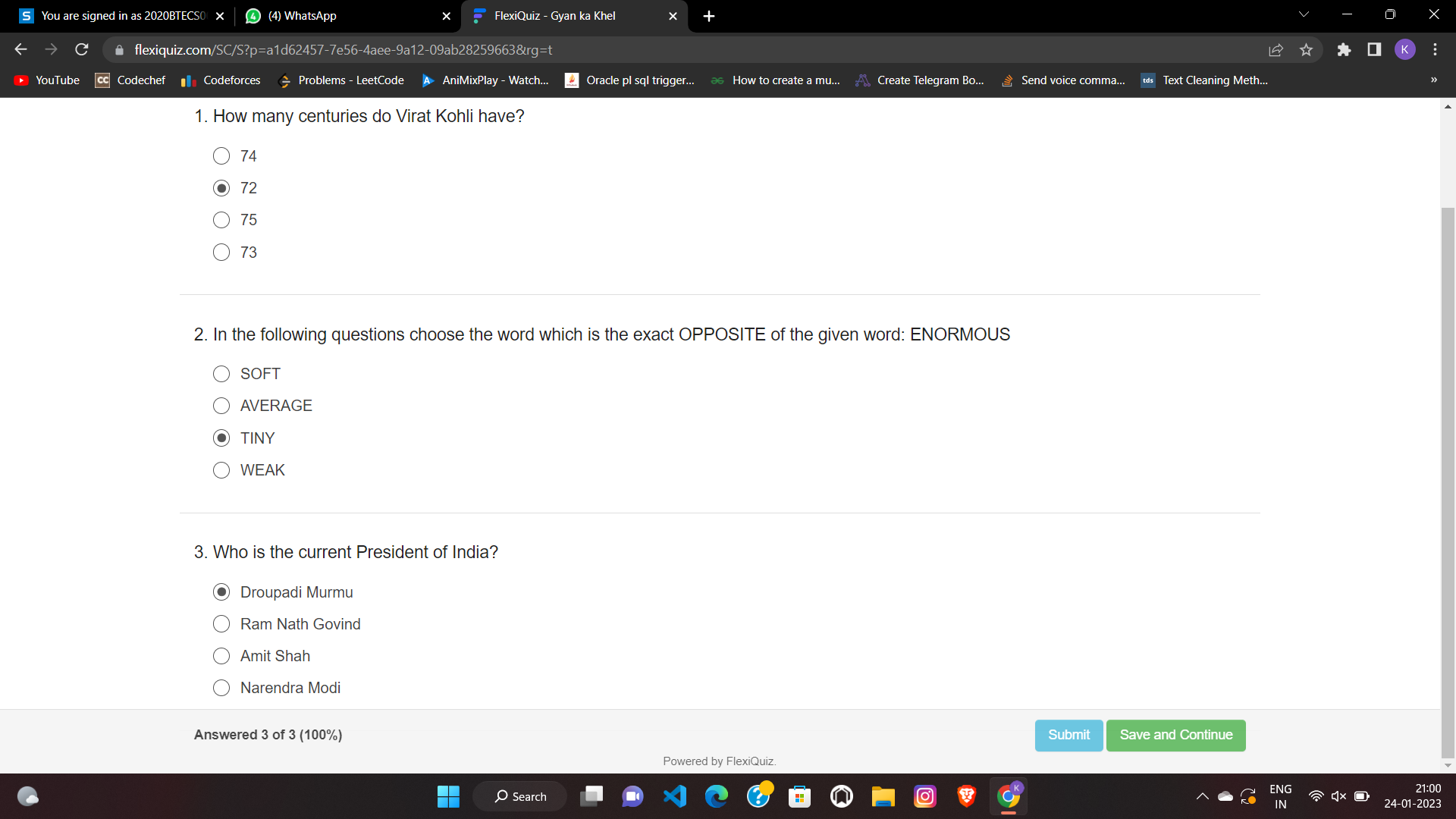
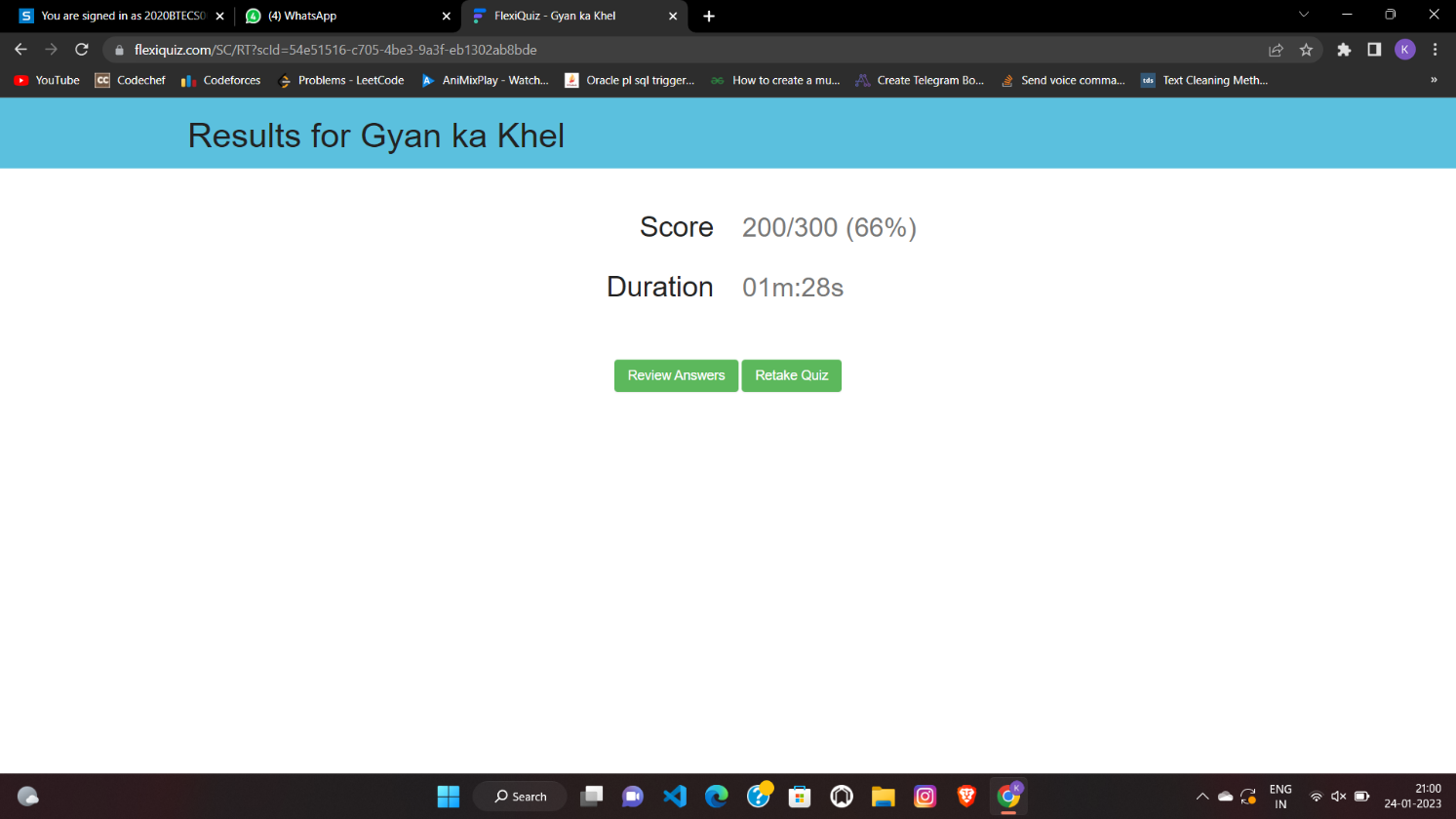
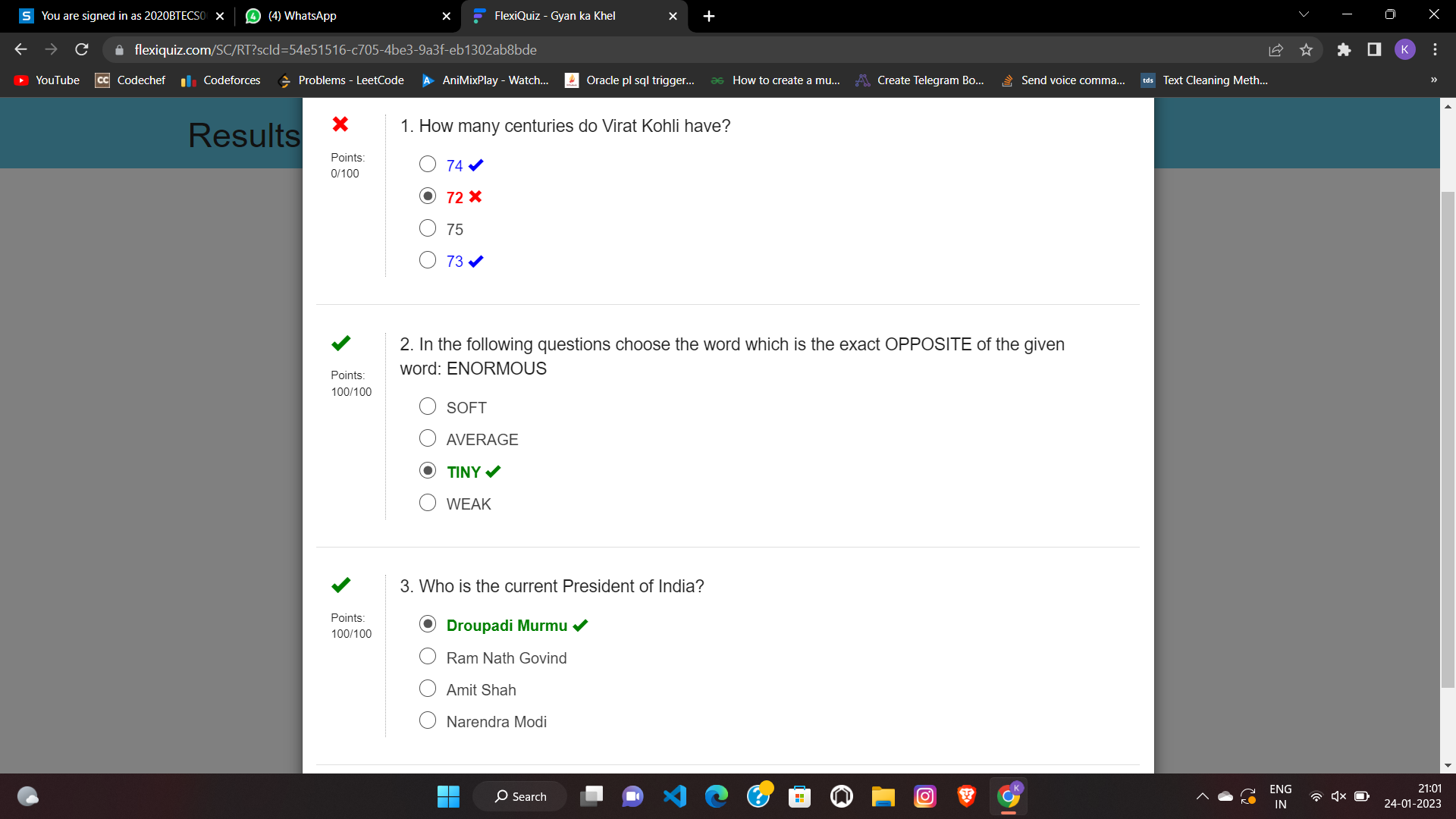
Mettl

FlexiQuiz

Eklavvya

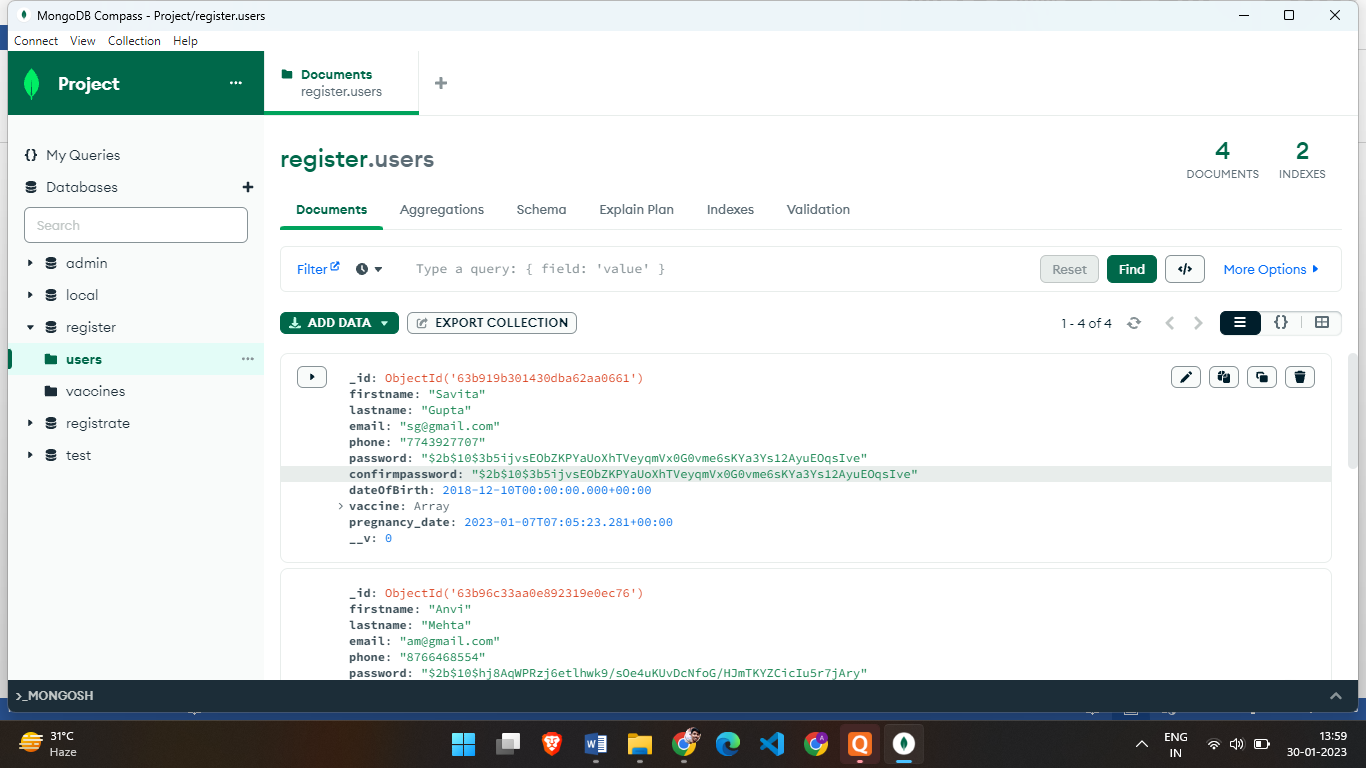
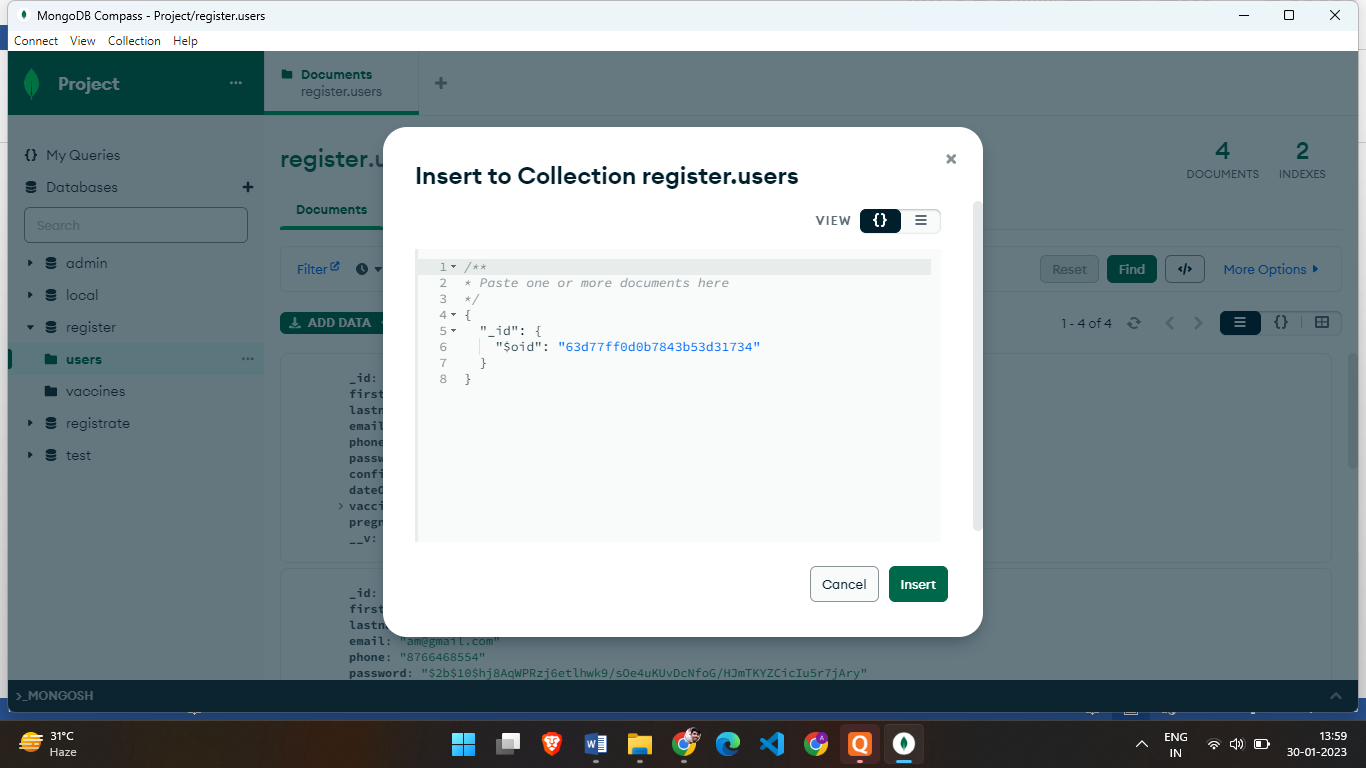
Think Exam

1. Demonstrate any one exam software which is open source and freely available.



1. Demonstrate FOSS software related to database.

* MongoDB:



1. How does the Exam software work?

* Exam Software works in the following steps:

Popularly used by recruiters and educational institutions, exam software is used for setting up online exams. The best online examination software helps with the following procedures:

Students’ Registration

Online exam software helps with the registration process of students and generates unique IDs for them.

Test Paper Creation

You can create a subjective, objective, multiple-choice, and other types of questions online and ensure zero spam.

Take Tests Anytime, Anywhere

Students can take tests from anywhere with a stable internet connection and a system. Similarly, teachers can invigilate directly through the system.

Automated Evaluation

Teachers don’t need to evaluate answers manually, as the exam software helps analyze students’ performance digitally.

Track Students’ Progress

YouTube broadcast software enables users to list their live streams as videos on their channels. This way the live stream can be seen even after it ended.

Data analysis

The performance reports include detailed info about the strengths and weaknesses of every student. Accordingly, teachers can make the improvement plan.