TITLE OF THE PROJECT

CODEXSPOT

ABSTRACT

The “CodexSpot” is a powerful software program designed to make learning easier, a response to the growing need for effective online learning environments. Platforms that can efficiently handle the wide range of educational interactions and information are desperately needed as the digital world changes and the need for remote learning increases. For instructors and students alike, CodexSpot acts as a central location that facilitates the easy storing, arrangement, and retrieval of a variety of course resources. CodexSpot offers an extensive feature set to improve the educational experience, ranging from course catalogs to instructor profiles, student enrollment to feedback aggregation.With the growing acceptance of online learning by both individuals and educational institutions, CodexSpot is prepared to tackle the issues of scalability, security, and user experience.

1. INTRODUCTION

Throughout the rapidly evolving field of online education, CodexSpot stands out as a trailblazing initiative with the goal of revolutionizing the educational process. There is an urgent need for platforms that can effectively curate and distribute educational content to a varied audience as education moves outside conventional boundaries.

CodexSpot recognizes the importance of accessibility in education. In a world where geographical barriers and resource limitations can hinder learning opportunities, CodexSpot strives to bridge these gaps by providing an inclusive platform accessible to anyone with an internet connection. Whether you're a student in a bustling metropolis or a remote village, CodexSpot ensures that high-quality educational resources are just a click away.

In order to meet this requirement CodexSpot provides a single library of courses covering a wide range of subjects and interests for students. With its easy-to-use interface and strong backend architecture, CodexSpot enables seamless interaction between educators and students when using educational resources. CodexSpot seeks to transform the exchange and consumption of knowledge in the digital era by promoting cooperation, creativity, and accessibility. CodexSpot, a leader in educational innovation, seeks to motivate and inspire a new generation of teachers and students by

2. AIMS AND OBJECTIVES

.

- The primary objective is to arrange data so that it can be retrieved with ease and structure. To do this, a schema representing the connections between various data elements must be designed.
- Make sure the Database is flexible and scalable to support future expansion an d changing student needs.
- ensure effective data storage and retrieval. To reduce response times, this involves improving query performance, indexing, and database design.
- To make sure the efficient functioning and maintenance of the database system, it is essential to document the database structure, data dictionary, and administer and user training.

3. HARDWARE /SOFTWARE REQUIREMENTS

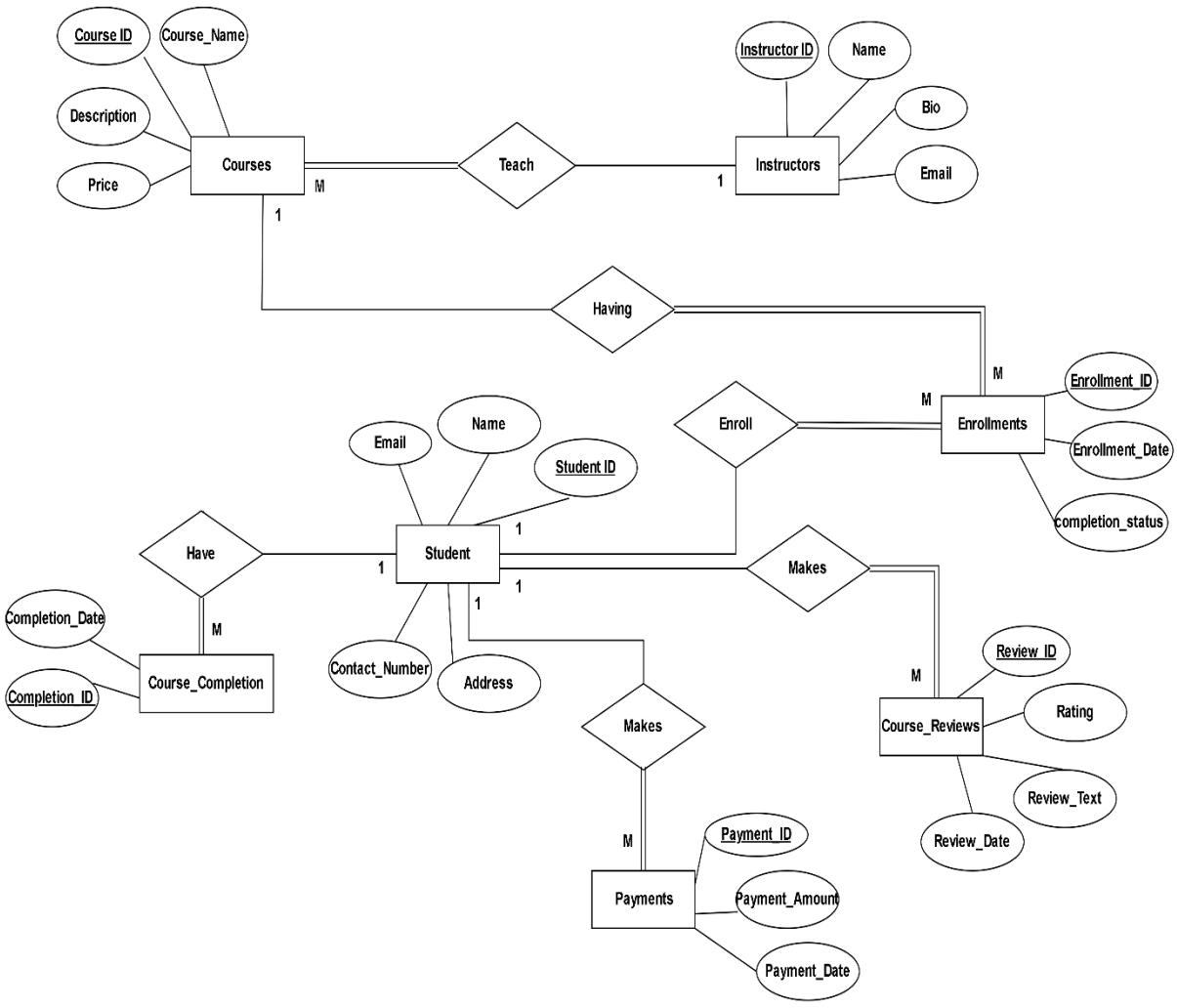
1. Operating System: Windows 7/8/10
2. Processor: 1.6 GHz or higher.
3. RAM: 2 GB or more.
4. Storage: 400 MB of available hard-disk space.
5. Graphics: A minimum display resolution of 1024x768 pixels.
6.software: MYSQL Workbench

4. DATABASE DESIGN

4.1 ER DIAGRAM

An Entity Relationship Diagram (ER diagram) shows entities, properties, and their relationships to visually represent the structure of a database. It makes complicated data models simpler, which helps with database design and comprehension. ER diagrams are crucial resources for clearly illustrating and explaining database topics

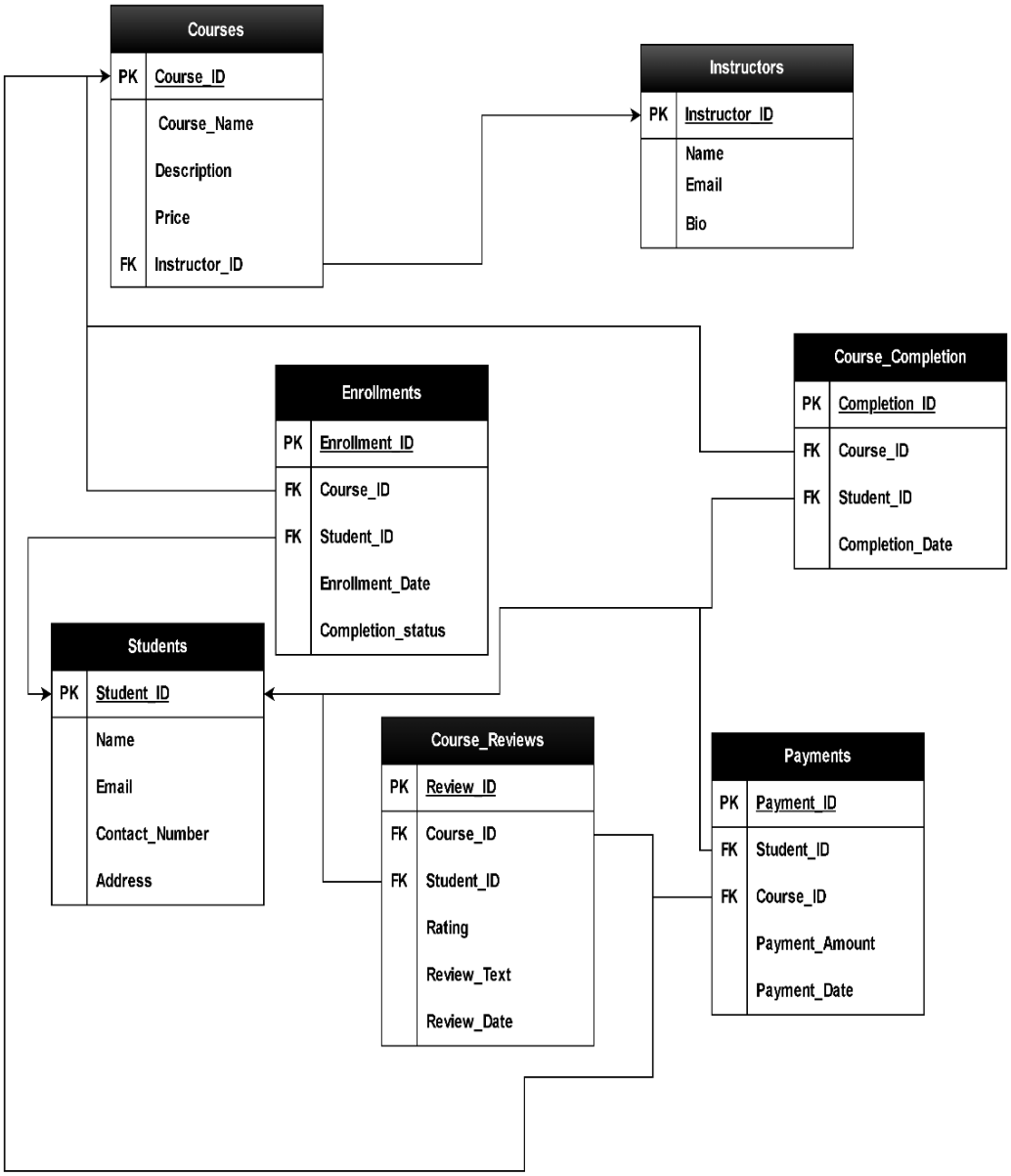
.



Fg-1

4.2 SCHEMA DIAGRAM

A schema diagram, also known as a database schema diagram, is a visual representation of the structure of a database. It illustrates the tables, columns, relationships, and constraints that make up the database schema



Fg-2

5. QUERIES AND RESULTS

1. Display list of all courses and their instructors, organized alphabetically by course names

```
SELECT c.Course_Name, i.Name AS Instructor_Name
FROM Courses c
INNER JOIN Instructors i ON c.Instructor_ID = i.Instructor_ID
ORDER BY c.Course_Name;
```

course_name	instrcutors_name
Ancient Civilizations	Dr. Michael Brown
Ecology and Conservation	Prof. Jennifer Martinez
Environmental Chemistry	Prof. William Martinez
Introduction to Programming	Dr. David Wilson
Introduction to Python	Prof. Emily Johnson
Java Programming	Prof. Sarah Davis
Machine learning	Dr. Elizabeth Taylor
Physics Fundamentals	Dr. John Smith
Sociology of Culture	Dr. Jessica Clark
Ux Designing	Prof. Robert Garcia

Result(1)

2. Display list of courses along with the count of enrolled students for each course Only courses with more than 1 enrollments.

```
SELECT c.course_name, COUNT(e.Student_ID) AS Enrolled_Students
FROM Courses c
LEFT JOIN Enrollments e ON c.Course_ID = e.Course_ID
GROUP BY c.course_id, c.course_name
HAVING COUNT(e.Student_ID) > 1;
```

course_name	enrolled_students
Introduction to Python	3
Java Programming	2
Machine learning	2

Result(2)

3. Display the list of students name along with the names of the courses they have completed

```
SELECT s.Name AS Student_Name, c.Course_Name AS Completed_Course
FROM Students s
INNER JOIN Course_Completion cc ON s.Student_ID = cc.Student_ID
INNER JOIN Courses c ON cc.Course_ID = c.Course_ID;
```

student_name	completed_course
Nithesh	Introduction to Python
Vickyath salian	Introduction to Python
Shreejesh Kumar	Introduction to Programming
Preethesh shetty	Java Programming
Nithesh	Introduction to Python
Vineeth G	Sociology of Culture
Sreeraj	Ecology and Conservation
Nithesh	Introduction to Python

Result(3)

4. Display the total payment for each courses, along with the course name

```
SELECT c.Course_Name, SUM(p.Payment_Amount) AS Total_Payment
FROM Payments p
JOIN Courses c ON p.Course_ID = c.Course_ID
GROUP BY c.Course_Name;
```

course_name	total_payments
Introduction to Python	4500.00
Java Programming	4000.00
Ecology and Conservation	500.00
Sociology of Culture	900.00
Introduction to Programming	1000.00
Machine learning	6000.00

Result(4)

5. Display the average rating for each course that has been reviewed

```
SELECT c.Course_Name, AVG(cr.Rating) AS Average_Rating
FROM Courses c
INNER JOIN Course_Reviews cr ON c.Course_ID = cr.Course_ID
GROUP BY c.Course_Name;
```

course_name	average_rating
Physics Fundamentals	5.0000
Introduction to Programming	3.5000
Ancient Civilizations	4.0000
Ecology and Conservation	4.0000

Result(5)

6. Display the latest enrollmentcourses with date, excluding courses that do not have any enrollments.

```
SELECT c.Course_Name, MAX(e.Enrollment_Date) AS
Latest_Enrollment_Date
FROM Courses c
JOIN Enrollments e ON c.Course_ID = e.Course_ID
GROUP BY c.Course_Name;
```

course_name	latest_enrollment_date
Introduction to Python	2024-04-20
Java Programming	2024-04-17
Ecology and Conservation	2024-04-15
Sociology of Culture	2024-04-14
Introduction to Programming	2024-04-13
Machine learning	2024-04-12

Result(6)

7. Create a trigger to automatically update the completion status in enrollments table when an an completion of course is done.

```
DELIMITER //
CREATE TRIGGER Completion_Trigger
AFTER INSERT ON Course_Completion
FOR EACH ROW
BEGIN
   UPDATE Enrollments
   SET Completion_Status = 'Completed'
   WHERE Student_ID = NEW.Student_ID AND Course_ID =
NEW.Course_ID;
END//
DELIMITER ;
```

Enrollment_ID	Student_ID	Course_ID	Enrollment_Date	Completion_Status
1	201	102	2024-04-20	Completed
2	202	102	2024-04-19	Not Completed
3	203	102	2024-04-18	Not Completed
4	204	104	2024-04-17	Not Completed
5	205	104	2024-04-16	Not Completed
6	206	106	2024-04-15	Not Completed
7	207	107	2024-04-14	completed
8	208	105	2024-04-13	Not Completed
9	209	109	2024-04-12	Not Completed
10	210	109	2024-04-11	Not Completed

Result(7)

8. Create a stored procedure to retrieve all the reviews for a given course.

```
DELIMITER //
CREATE PROCEDURE GetCourseReviews(courseID INT)
BEGIN
   SELECT r.Rating, r.Review_Text, r.Review_Date, s.Name AS
Student_Name
   FROM Course_Reviews r
   JOIN Students s ON r.Student_ID = s.Student_ID
   WHERE r.Course_ID = courseID;
END //

DELIMITER ;
```

```
mysql> call getcoursereviews(101);
+--------+-----------------------------------------------------------------------------------+-------------+-----------------+
| Rating | Review_Text                                                                       | Review_Date | Student_Name    |
+--------+-----------------------------------------------------------------------------------+-------------+-----------------+
|      5 | Fantastic course! Dr. Smith explains complex concepts in an easy-to-understand manner. | 2024-04-25  | Nithesh         |
|      5 | Enjoyed the course, learned a lot                                                 | 2024-05-13  | Shreejesh Kumar |
+--------+-----------------------------------------------------------------------------------+-------------+-----------------+
```

Result(8)

9. Create a view that displays details about course enrollments, including student names and contact information, along with course names and instructor names

```
CREATE VIEW CourseEnrollmentsView  AS
SELECT e.Enrollment_ID, s.Name AS Student_Name, s.Email AS
Student_Email, s.Contact_Number AS Student_Contact,
    c.Course_Name,  c.Price, i.Name AS Instructor_Name
FROM Enrollments e
JOIN Students s ON e.Student_ID = s.Student_ID
JOIN Courses c ON e.Course_ID = c.Course_ID
JOIN Instructors i ON c.Instructor_ID = i.Instructor_ID;
```

```
mysql> select * from courseEnrollmentsview;
+---------------+-----------------+---------------------+----------------+---------------------------+---------+-------------------------+
| Enrollment_ID | Student_Name    | Student_Email       | Student_Contact | Course_Name               | Price   | Instructor_Name         |
+---------------+-----------------+---------------------+----------------+---------------------------+---------+-------------------------+
|             1 | Nithesh         | nithu@example.com   | 7909309845     | Introduction to Python    | 1500.00 | Prof. Emily Johnson     |
|             2 | Rakshith rai A  | rakshith@example.com | 9876543210     | Introduction to Python    | 1500.00 | Prof. Emily Johnson     |
|             3 | Vickyath salian | vicky@example.com   | 8567891234     | Introduction to Python    | 1500.00 | Prof. Emily Johnson     |
|             4 | Sreeraj         | raj@example.com     | 9216549870     | Java Programming          | 2000.00 | Prof. Sarah Davis       |
|             5 | Preethesh shetty | preethu@example.com | 7891234567     | Java Programming          | 2000.00 | Prof. Sarah Davis       |
|             6 | Bhooshan A      | bhooshan@example.com | 7564893546     | Ecology and Conservation  |  500.00 | Prof. Jennifer Martinez |
|             7 | Vineeth G       | vinni@exmaple.com   | 7903456376     | Sociology of Culture      |  900.00 | Dr. Jessica Clark       |
|             8 | Shreejesh Kumar | shreeju@example.com | 7695453678     | Introduction to Programming | 1000.00 | Dr. David Wilson        |
|             9 | Sonith Sk       | soni@example.com    | 9826345647     | Machine learning          | 3000.00 | Dr. Elizabeth Taylor    |
|            10 | Maneesh         | maneesh@example.com | 7834562345     | Machine learning          | 3000.00 | Dr. Elizabeth Taylor    |
+---------------+-----------------+---------------------+----------------+---------------------------+---------+-------------------------+
```

Result(9)

6.CONCLUSION

In conclusion ,CodexSpot, which combines a user-friendly interface with a strong database backend, is a prime example of how online education is changing. Our investigation has revealed details about its design and operation, highlighting how well it handles course materials, student communications, and financial transactions. Even if CodexSpot shows a lot of promise, there is still opportunity for improvement, especially in areas like adaptive course recommendations, personalized learning experiences, and the incorporation of cutting-edge technology like VR and AI.

7.FUTURE ENHANCEMENT

- Development of mobile application
- Implementing personalized learning paths based on student preferences, learning styles, and performance metrics.
- Enhancing interactivity through live virtual classrooms, interactive simulations, and gamified learning experiences.

8.REFERENCES

1)C. J. Date, A. Kannan and S. Swamynathan, An Introduction to Database Systems , Pearson Education, Eighth Edition, 2009.

2)Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database systems concepts,McGraw-Hill Education (Asia), Fifth Edition, 2006.

3)Shio Kumar Singh, database system Concepts,,Designs and Application,Person Education,Second Edition 2011.

4)Peter Rob and Carlos Coronel, Database Systems Designs Implementations and Management,Thomson Learning-Course Technology ,Seventh Edition, 2007.