**SCRIPT EVALUATION ASSISTANT**

**A PROJECT REPORT**

***Submitted by***

N.AJAY RAJ 316126510034

M.RAGHU 316126510031

N.AMRUTHA LAKSHMI 316126510032

M.SUSHMA 316126510027

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**



Under esteemed guidance of

**Mrs.Keerthi Lingam**

(Assistant Professor)

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Affiliated to Andhra University)

**SANGIVALASA, VISAKHAPATNAM - 531162**

2017-2018

**BONAFIDE CERTIFICATE**

Certified that this project report “Script Evaluation Assistant” is the bonafide work of N. Ajay Raj (316126510034), M. Raghu (316126510031), N. Amrutha Lakshmi (316126510032) and M. Sushma (316126510027) who carried out the project work under my supervision.

DECLARATION

This is to certify that the project work entitled “**SCRIPT EVALUATION ASSISTANT**” is a bonafide work carried out by **N.AJAY RAJ, M.RAGHU, N.AMRUTHA LAKSHMI, M.SUSHMA** as a part of **B.TECH** third year 1st semester of **Computer Science Engineering** of ANITS, Visakhapatnam during the year 2018-19.

We, **N.AJAY RAJ, M.RAGHU, N.AMRUTHA LAKSHMI, M.SUSHMA** of third year B.Tech., in the department of Computer Science Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled  **SCRIPT EVALUATION ASSISTANT** is carried out by us and submitted in partial fulfilment of the requirements for the award of **Bachelor of Technology in Computer Science Engineering,** under Anil Neerukonda Institute of Technology & Sciences during the academic year 2018-19 and has not been submitted to any other university for the award of any kind of degree.

N.AJAY RAJ 316126510034

M.RAGHU 316126510031

N.AMRUTHA LAKSHMI 316126510032

M.SUSHMA 316126510027

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement always boosted the morale. We take a great pleasure in presenting a project, which is the result of a studied blend of both research and knowledge.

We first take the privilege to thank the Head of our Department, **Dr.R.SIVARANJANI**, for permitting us in laying the first stone of success and providing the lab facilities, we would also like to thank the other staff in our department and lab assistants who directly or indirectly helped us in successful completion of the project.

We feel great to thank **Mrs.KEERTHI LINGAM** who is our project guide and who shared her valuable knowledge with us and made us understand the real essence of the topic and created interest in us to work day and night for the project; we also thank our B.Tech project coordinators **Mr.K.SURESH**  for their support and encouragement.

N.AJAY RAJ 316126510034

M.RAGHU 316126510031

N.AMRUTHA LAKSHMI 316126510032

M.SUSHMA 316126510027

**ABSTRACT**

SCRIPT EVALUATION ASSISTANT is a software system, which allows faculty to collect, evaluate and manage grades. The main purpose of this study was to provide a runtime environment for instructors and faculty to execute and evaluate the students performance and store them in database for later use . The tools used are Python programming, and Tkinter for front end purpose.

TABLE OF CONTENTS

1. Introduction
   1. Problem Statement
   2. Contribution
2. Domain Explanation
3. Relations
4. ER Diagram
5. Integrity constraints and referential integrity constraints
6. Tables description
7. Schema diagram
8. Queries
9. Screenshots and navigation of pages
10. Conclusion
11. INTRODUCTION

1.1 PROBLEM STATEMENT

It is an usual practice in Engineering colleges to grade students lab observations and records for internal assessment. This project provides a runtime for the faculty to run the students program and grade them accordingly. This grading and file locations will be stored in the database which can be used to generate spreadsheets and have an electronic record management system rather than using paper and manual process.

1.2 CONTRIBUTION

The main purpose of project is to award marks for students assignment submission according to their date of submission. It makes the work uncomplicated for faculty while reduces their manual work and improves accuracy.

2. DOMAIN EXPLANATION

**2.1 SYSTEM INTERFACE**:

Application would be a self contained system.It will not access the data of any other application that have accessed to its data.

2.2 USER INTERFACE:

Application would be accessed by a browser interface.The interface would be viewed best using 1366x768 and 1440x900 pixels resolution setting.The software would be fully compatible with Tkinter.

2.3 HARDWRE INTERFACE:

Server: Faculty laptop, 4GB RAM, core i3 and above

Client: Student PC with their run time, 2GB RAM, Network Interface Card

2.4 SOFTWARE INTERFACE:

Python 3.6 and above, mysql 8.0 and above, Google Drive Client

2.5 COMMUNICATION INTERFACE:

The system should be accessed over LAN or WAN. For clients to access the application server. Network should be running on TCP/IP protocol.

2.6 OPERATIONS:

Client Side: Students will be able to upload program scripts.

Server Side: Run programs, evaluate and generate spread sheets.

3. RELATIONS

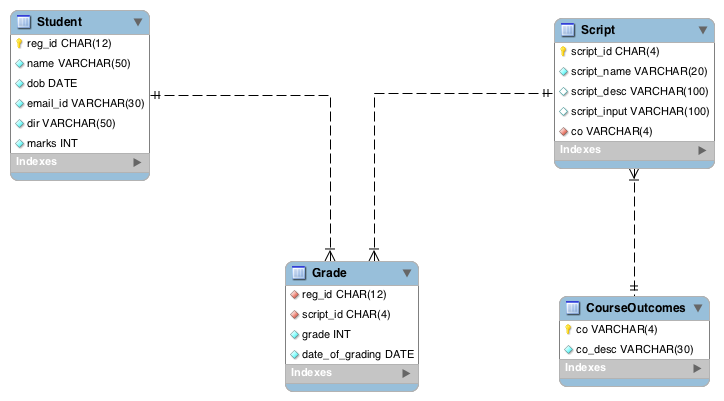
Student(reg\_id:char, name:varchar, dob:date, email\_id:varchar, dir:varchar, marks:integer);

Script(script\_id:char, script\_name:varchar, script\_desc:varchar, script\_input:varchar, co:varchar);

Grade(reg\_id:char, script\_id:char, grade:integer, date\_of\_grading:date);

CourseOutcomes(co:varchar, co\_desc:varchar);

4. SCHEMA



5. INTEGRITY CONSTRAINTS AND REFERENTIAL INTEGRITY CONSTRAINTS

Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. The whole purpose of constraints is to maintain the **data integrity**during an update/delete/insert into a table. In this tutorial we will learn several types of constraints that can be created in RDBMS.

## Types of constraints

* NOT NULL
* UNIQUE
* DEFAULT
* CHECK
* Key Constraints – PRIMARY KEY, FOREIGN KEY

#### NOT NULL:

NOT NULL constraint makes sure that a column does not hold NULL value. When we don’t provide value for a particular column while inserting a record into a table, it takes NULL value by default. By specifying NULL constraint, we can be sure that a particular column(s) cannot have NULL values.

#### UNIQUE:

UNIQUE Constraint enforces a column or set of columns to have unique values. If a column has a unique constraint, it means that particular column cannot have duplicate values in a table.

#### DEFAULT:

The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

#### CHECK:

This constraint is used for specifying range of values for a particular column of a table. When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

## Key constraints:

#### PRIMARY KEY:

Primary key uniquely identifies each record in a table. It must have unique values and cannot contain nulls. In the below example the ROLL\_NO field is marked as primary key, that means the ROLL\_NO field cannot have duplicate and null values.

#### FOREIGN KEY:

Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

6. TABLES DESCRIPTION

-- Creating Database --

create database SEA;

-- Using/naviagte to Sea database --

use SEA;

-- Creating Student Table --

create table Student (

reg\_id char(12) primary key,

name varchar(50) not null,

dob date not null,

email\_id varchar(30) not null,

dir varchar(50) not null,

marks integer not null

);

-- Creating Script Table --

create table Script (

script\_id char(4) primary key,

script\_name varchar(20) unique not null,

script\_desc varchar(100),

script\_input varchar(100),

co varchar(4) not null

);

-- Creating Grading Table --

create table Grade (

reg\_id char(12) not null,

script\_id char(4) not null,

grade integer not null,

date\_of\_grading date not null

);

-- adding reg\_id constraint --

alter table Grade

add constraint regid\_fk foreign key(reg\_id)

references Student(reg\_id);

-- adding script\_id constraint --

alter table Grade

add constraint scriptid\_fk foreign key(script\_id)

references Script(script\_id);

-- adding course outcomes table --

create table CourseOutcomes (

co varchar(4) primary key,

co\_desc varchar(30) not null

);

-- adding co constraint --

alter table Script

add constraint co\_fk foreign key(co)

references CourseOutcomes(co);