VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590018, Karnataka

INTERNSHIP REPORT

ON

"CGPA Calculator-VTU Site"

Submitted in partial fulfilment for the award of degree(21CSI85)

BACHELOR OF ENGINEERING IN
YOUR BRANCH

Submitted by:

SURBHI KUMARI

1CD20AI044

Conducted at
VARCONS TECHNOLOGIES

CAMBRIDGE INSTITUTE OF TECHNOLOGY

Department of Artificial Intelligence And Machine Learning

Accredited by NBA, New Delhi

Main Campus KR Puíam, Bengaluíu - 5G003G

CAMBRIDGE INSTITUTE OF TECHNOLOGY

Department of Artificial Intelligence And Machine Learning

Accredited by NBA, New Delhi

Main Campus KR Puíam, Bengaluíu - 5G003G

CERTIFICATE

This is to certify that the Internship titled **"CGPA Calculator-VTU Site"** carried out by **Ms .Surbhi Kumari,** a bonafide student of Cambridge Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering**, in **Artificial Intelligence and Machine Learning** under Visvesvaraya Technological University, Belagavi, during the year 2023-2024. It is certified that all corrections/suggestions indicated have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements in respect of Internship prescribed for the course Internship / Professional Practice (21****)

Signature of Guide **Signature of HOD** **Signature of Principal**

External Viva:

Name of the Examiner Signature with Date

1)_____

_____

2)_____

_____

D E C L A R A T I O N

I, **Surbhi Kumari**, final year student of AIML, Cambridge Institute of Technology -560049, declare that the Internship has been successfully completed, in **VARCONS TECHNOLOGIES**. This report is submitted in partial fulfillment of the requirements for award of Bachelor Degree in Artificial Intelligence and Machine Learning, during the academic year 2023-2024.

Date :18-04-2024 :

Place : Bengaluru

USN : 1CD20AI044

NAME: Surbhi Kumari

Internship Offer Letter

Date: 5th March, 2024

Name: **Surbhi kumari**
USN: **1CD20AI044**
Placement ID: **FSWDMARCHBATCHONE**

Dear Student,

We would like to congratulate you on being selected for the **Full Stack Web Devlopment** Internship position with **Varcons Technologies**, effective Start Date **5th March, 2024**, All of us are excited about this opportunity provided to you!

This internship is viewed as being an educational opportunity for you, rather than a part-time job. As such, your internship will include training/orientation and focus primarily on learning and developing new skills and gaining a deeper understanding of concepts of **Full Stack Web Devlopment** through hands-on application of the knowledge you learn while you train with the senior developers. You will be bound to follow the rules and regulations of the company during your internship duration.

Again, congratulations and we look forward to working with you!.

Sincerely,

Spoorthi H C
Director
Varcons Technologies
213, 2st Floor, 18 M G Road, Ulsoor,
Bangalore-560001

OFFER LETTER PROVIDED BY THE COMPANY

A C K N O W L E D G E M E N T

This Internship is a result of accumulated guidance, direction and support of several important persons. We take this opportunity to express our gratitude to all who have helped us to complete the Internship.

We express our sincere thanks to our Principal, for providing usadequate facilities to undertake this Internship.

We would like to thank our Head of Dept – AIML, for providing us an opportunity to carry out Internship and for his valuable guidance and support.

We would like to thank our (Lab assistant name) Software Services for guiding us during the period of internship.

We express our deep and profound gratitude to our guide, Prof. Geetha R, Assistant Prof, for her keen interest and encouragement at every step in completing the Internship.

We would like to thank all the faculty members of our department for the support extended during the course of Internship.

We would like to thank the non-teaching members of our dept, for helping us during the Internship.

Last but not the least, we would like to thank our parents and friends without whose constant help, the completion of Internship would have not been possible.

Surbhi Kumari
1CD20AI044

ABSTRACT

This internship report presents an analysis of the CGPA (Cumulative Grade Point Average) calculator tool integrated into the Visvesvaraya Technological University (VTU) website, with a focus on enhancing user experience. The report outlines the methodologies employed to evaluate the effectiveness, usability, and functionality of the CGPA calculator tool.

The study begins with a comprehensive review of existing CGPA calculators and user interface design principles. Subsequently, various metrics are identified to assess the usability and user satisfaction of the VTU CGPA calculator. These metrics include efficiency, effectiveness, learnability, error prevention, and user satisfaction.

To gather empirical data, a combination of quantitative and qualitative research methods is employed. Quantitative data is collected through user surveys and analytics tools, while qualitative insights are derived from user interviews and usability testing sessions. The analysis of collected data provides valuable insights into user preferences, pain points, and suggestions for improvement.

The findings of this study highlight several key areas for enhancement in the VTU CGPA calculator, including interface design adjustments, feature additions, and performance optimization. Additionally, recommendations are provided for implementing these improvements to ensure a more intuitive and seamless user experience. Ultimately, this internship report aims to contribute to the ongoing efforts of VTU in enhancing its online services, particularly in the realm of academic tools, by providing actionable insights derived from user-centered research and analysis.

The successful implementation of the CGPA calculator will contribute to streamlining the academic processes at VTU, empowering students to track their progress and make informed decisions regarding their educational journey. Additionally, it will demonstrate the university's commitment to leveraging technology to enhance the student experience and foster academic excellence.

Table of Contents

CHAPTER 1

COMPANY PROFILE

1. COMPANY PROFILE

A Brief History of Company

Varcons Technologies is a leading provider of cutting-edge technologies and services, offering scalable solutions for businesses of all sizes. Founded by a group of friends who started by scribbling their ideas on a piece of paper, today we offer smart, innovative services to dozens of clients. We develop SaaS products, provide Corporate Seminars, Industrial trainings and much more.

Smart solutions are at the core of all that we do at VCT. Our main goal is to find smart ways of using technology that will help build a better tomorrow for everyone, everywhere. SaaS offers a variety of advantages over traditional software licensing models and We here at VCT tend to include the key features of SaaS in everything we build.

We develop websites that behave and interact similar to Sophisticated software and also provied Information+Functionality=WaaS .Let us analyse the way your users/customers interact with you/your business by gathering, studying, and understanding the consumer voice and their perception of the product/service to generate a report to help you make better market decisions.With a comprehensive range of services, We can guarantee your technology needs are not just met, but exceeded. We shall work with your Customers/users closely to understand the way your users/customers use/make use of Products/Services. We create API's and tools that help you automate any process with a host of features pertaining to the Device.

VCA provides a host of services to its customers/users/clients, Enabling business success driven by technology Harnessing the power of technology, we create a measurable difference for our clients across various industries & multiple geographies. We develop websites that behave and interact similar to sophisticated software. We help you manage your SEO campaign more efficiently and effectively. We help you gain market share by leveraging our expertise. our holistic approach to identify anything that may be hurting your traffic or rankings and show you just how to outrank the competition.

CHAPTER 2

ABOUT THE COMPANY

2. ABOUT THE COMPANY

Built for Creatives, by Creatives at VCT, We make sure every product/service that we offer is built keeping in mind the practical usability of the product/Service, We're a startup focused on Creativity and Customizability, and We also provide subscription models for Software that we have already built, Since the application is already configured, the user has a ready-to-use application. This not only reduces installation and configuration time but also cuts down the time wasted on potential glitches linked to software deployment.

Departments and services offered

VCA provides a host of services to its customers/users/clients, Enabling business success driven by technology Harnessing the power of technology, we create a measurable difference for our clients across various industries & multiple geographies. We develop websites that behave and interact similar to sophisticated software. We help you manage your SEO campaign more efficiently and effectively. We help you gain market share by leveraging our expertise. our holistic approach to identify anything that may be hurting your traffic or rankings and show you just how to outrank the competition.

With a comprehensive range of services, we guarantee your technology needs are not just met, but exceeded. We shall work with your customers/users closely to understand the way your users/customers use/make use of products/services We offer professional Graphic design, Brochure design & Logo design. We are experts in crafting visual content to convey the right message to the customers.We also design custom wraps for your products(also known as package designing) We analyse the way your users/customers interact with you/your business by gathering, studying and understanding the consumer voice and their perception of the product/service We work with Consumer Electronics, Lighting, Home Automation, Metering, Sensor-Technology, Home Appliance and Medical Device companies to help them create smart and connected products. Through its integrated Embedded and IoT services, VCA helps build intelligent & connected devices that can be remotely monitored and controlled while leveraging edge and cloud computing for a host of intelligent applications and analytics.

Services provided by Company.

- Website as Software
- Search Engine Optimisation
- Comprehensive Customer Support
- Branding and Design
- Analytics and Research
- Embedded Systems and IOT

OD services we provide: Traditional Services+ SaaS features= Magic!

- Website as Software
- Analytics and Research
- Comprehensive Customer Support
- Smart Automation Tools

We develop websites that behave and interact similar to Sophisticated software. Information+Functionality=WaaS .Let us analyse the way your users/customers interact with you/your business by gathering, studying, and understanding the consumer voice and their perception of the product/service to generate a report to help you make better market decisions.
With a comprehensive range of services, We can guarantee your technology needs are not just met, but exceeded. We shall work with your Customers/users closely to understand the way your users/customers use/make use of Products/Services. We create API's and tools that help you automate any process with a host of features pertaining to the Device.

CHAPTER 3

INTRODUCTION

3. INTRODUCTION

Introduction to CGPA Calculator-VTU Site

The modern educational landscape is increasingly reliant on digital tools and platforms to streamline administrative processes and enhance the learning experience. In this context, the development of a CGPA (Cumulative Grade Point Average) calculator for the Visvesvaraya Technological University (VTU) website emerges as a significant endeavor aimed at facilitating student academic management and improving user interaction with the university's online resources.

The proposed project seeks to address this challenge by designing and implementing a user-friendly CGPA calculator seamlessly integrated into the VTU website. By leveraging modern web development technologies and adhering to user-centric design principles, the calculator aims to provide students with a convenient and accessible tool for computing their CGPA accurately.

Key features of the CGPA calculator include:

- User-friendly interface: The calculator will feature an intuitive interface designed for ease of use, allowing students to input their grades and credits effortlessly.
- Accurate computation: The calculator will employ algorithms to accurately calculate the CGPA based on the grades obtained by the student in each course and the corresponding credit weightage.
- Flexibility: The calculator will accommodate various grading systems used by VTU across different programs and semesters, ensuring flexibility and accuracy in CGPA calculation.
- Interactive feedback: The calculator will provide instant feedback to users, allowing them to review and adjust their inputs to obtain the desired CGPA.
- Accessibility: The calculator will be accessible to all VTU students through the official website, ensuring equitable access to this essential academic tool.

Problem Statement

The absence of a dedicated Cumulative Grade Point Average (CGPA) calculator on the Visvesvaraya Technological University (VTU) website poses a significant challenge for students in efficiently computing and tracking their academic performance. This project aims to address this issue by developing and integrating a user-friendly CGPA calculator into the VTU website, providing students with a convenient and accessible tool to accurately calculate and monitor their CGPA, thereby enhancing the overall student experience and academic management within the university.

CHAPTER 4

SYSTEM ANALYSIS

4. SYSTEM ANALYSIS

1. System Analysis

A CGPA (Cumulative Grade Point Average) calculator is a vital tool for students and academic institutions alike, facilitating the computation of a student's overall academic performance. Effective system analysis is crucial in designing a CGPA calculator that meets the needs of users while ensuring accuracy, efficiency, and usability. This analysis delves into the process of identifying requirements, designing system architecture, and outlining development and deployment strategies for a CGPA calculator.

User Requirements Gathering:

Understanding user needs is paramount in developing a successful CGPA calculator. Through interviews and surveys with students, academic advisors, and administrators, key features and functionalities are identified. Users express a desire for a user-friendly interface allowing easy input of grades, calculation of CGPA, visualization of semester-wise performance, and generation of comprehensive reports.

Functional Requirements Identification:

The core functions of the CGPA calculator are delineated, encompassing grade input, CGPA computation, secure data storage, and report generation. Additional features such as grade conversions and support for various grading systems are specified to enhance user experience and accommodate diverse academic contexts.

Non-functional Requirements Determination:

Non-functional requirements are defined to ensure the system's performance, usability, security, and reliability. Performance considerations encompass fast response times and scalability to accommodate increasing user loads. Usability is prioritized through an intuitive interface suitable for users with varying technical proficiencies. Security measures include robust data encryption and user authentication mechanisms, while reliability entails error handling and fault tolerance.

System Architecture Design:

The system architecture is designed to accommodate scalability, maintainability, and compatibility. Modular components are established for the user interface, data processing, and storage, allowing for flexibility and ease of future

enhancements or modifications. Technologies and frameworks are chosen based on their suitability for the project's requirements and constraints.

Data Modeling:

A structured database schema is designed to efficiently store student information, including personal details, course enrollment, grades, and CGPA calculations. Relationships between entities such as students, courses, and semesters are established to ensure data integrity and consistency.

Interface Design

The user interface is meticulously crafted to optimize usability and accessibility. Wireframes and mockups are created, incorporating stakeholder feedback to design an intuitive input form for grades and interactive visualizations for CGPA trends and performance metrics.

Development and Testing:

System development follows best practices for coding, documentation, and version control. Rigorous testing, including unit tests, integration tests, and user acceptance tests, is conducted to identify and rectify bugs or inconsistencies before deployment.

Deployment and Maintenance:

The CGPA calculator is deployed in a production environment, seamlessly integrating with existing IT infrastructure. Ongoing maintenance and support are provided to address issues, implement new features, and adapt to evolving user needs and technological advancements.

2. Objective of the System

The CGPA Calculator system aims to streamline the process of computing Cumulative Grade Point Averages (CGPA) by providing students, academic advisors, and administrators with a reliable and user-friendly platform. Its primary objective lies in ensuring accuracy in CGPA calculation, leveraging grades obtained in individual courses and accounting for their respective credit weightage. Efficiency is another core objective, with the system designed to swiftly process grade inputs and generate comprehensive reports, thereby minimizing the time and effort required for academic analysis. User-centric design principles are integral, with an intuitive interface facilitating easy grade input, CGPA visualization, and access to insightful performance analytics. Flexibility is emphasized to accommodate diverse grading systems and formats, ensuring the system's applicability across various educational contexts.

In addition to accuracy and efficiency, the CGPA Calculator system prioritizes data security, implementing robust encryption and authentication mechanisms to safeguard sensitive student

information. The system's architecture is designed for scalability, capable of accommodating increasing user loads and future enhancements while ensuring consistent reliability and minimal downtime. Insightful analytics provide stakeholders with valuable insights into academic performance, including semester-wise trends and areas for improvement, contributing to informed decision-making and academic planning. Ultimately, the system aims to enhance stakeholder satisfaction by meeting the diverse needs and expectations of students, faculty, and academic administrators, thereby fostering a conducive environment for academic success and progression.

CHAPTER 5

REQUIREMENT ANALYSIS

5. REQUIREMENT ANALYSIS

Hardware Requirement Specification

- MySQL
- NODE JS
- Notepad++ Editor
- Processor: Intel core i5 processer
- Memory: 15.6 GB
- Hard Disk: 40 GB

Software Requirement Specification

A] Functional Requirements

- Python
- CSS
- HTML.

B] Non-Functional Requirements

- **Availability**

The online registration system shall permit backing up of the registration database whileother registration actives are going on.

- **Accessibility**

The system shall be accessible by people with specific vision needs to the extent that a user shall be able to display whole user interface in a larger font without truncating displayed text or other values.

- **Security**

The access permissions for system data may only be change by the systems data administrator passwords shall never be viewable at the point of entry or any other time.

CHAPTER 6

DESIGN ANALYSIS

6. DESIGN & ANALYSIS

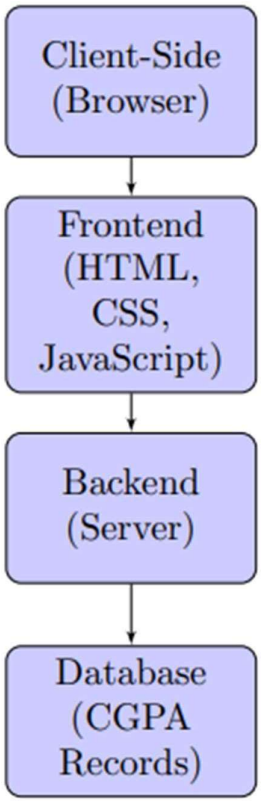


Fig System Design

CHAPTER 7

IMPLEMENTATION

7. IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and it constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning.

Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

TESTING

The testing phase is an important part of software development. It is the Information zed system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Software testing is carried out in three steps:

1. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether theobjectives have been met. Errors are noted down and corrected immediately.
2. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.
3. The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole.

```
from PyQt5 import QtCore, QtGui, QtWidgets
import sys
import webbrowser
import os
grade_points = []
credits = []
total_marks = []
lcdNumber = True
lcdNumber_1 = True
credits_earned = []
cixgi = []
ci_earned =[]

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(957, 889)
        MainWindow.setMinimumSize(QtCore.QSize(740, 650))
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap("ico.ico"), QtGui.QIcon.Normal, QtGui.QIcon.Off)
        MainWindow.setWindowIcon(icon)
        MainWindow.setStyleSheet("background-color: rgb(40, 42, 54);\n"
"")
        MainWindow.setTabShape(QtWidgets.QTabWidget.Rounded)
        MainWindow.setUnifiedTitleAndToolBarOnMac(True)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.tabWidget = QtWidgets.QTabWidget(self.centralwidget)
        self.tabWidget.setGeometry(QtCore.QRect(0, 0, 961, 891))
        self.tabWidget.setStyleSheet("background-color: rgb(170, 255, 0);")
        self.tabWidget.setTabPosition(QtWidgets.QTabWidget.North)
        self.tabWidget.setTabShape(QtWidgets.QTabWidget.Rounded)
        self.tabWidget.setIconSize(QtCore.QSize(20, 20))
        self.tabWidget.setElideMode(QtCore.Qt.ElideNone)
        self.tabWidget.setTabsClosable(False)
        self.tabWidget.setMovable(False)
        self.tabWidget.setTabBarAutoHide(False)
        self.tabWidget.setObjectName("tabWidget")
        self.sgpa_with_grades = QtWidgets.QWidget()
        self.sgpa_with_grades.setStyleSheet("background-color: rgb(153, 155, 255);")
        self.sgpa_with_grades.setObjectName("sgpa_with_grades")
        self.gridLayoutWidget = QtWidgets.QWidget(self.sgpa_with_grades)
        self.gridLayoutWidget.setGeometry(QtCore.QRect(10, 50, 921, 421))
        self.gridLayoutWidget.setObjectName("gridLayoutWidget")
        self.gridLayout = QtWidgets.QGridLayout(self.gridLayoutWidget)
        self.gridLayout.setContentsMargins(0, 0, 0, 0)
        self.gridLayout.setObjectName("gridLayout")
        self.comboBox_3 = QtWidgets.QComboBox(self.gridLayoutWidget)
        self.comboBox_3.setObjectName("comboBox_3")
        self.comboBox_3.addItem("")
        self.comboBox_3.addItem("")
```

```
self.comboBox_3.addItem("")
self.comboBox_3.addItem("")
self.comboBox_3.addItem("")
self.gridLayout.addWidget(self.comboBox_3, 2, 1, 1, 1)
self.label_10 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_10.setObjectName("label_10")
self.gridLayout.addWidget(self.label_10, 5, 0, 1, 1)
self.spinBox_7 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_7.setObjectName("spinBox_7")
self.gridLayout.addWidget(self.spinBox_7, 7, 2, 1, 1)
self.spinBox_4 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_4.setObjectName("spinBox_4")
self.gridLayout.addWidget(self.spinBox_4, 4, 2, 1, 1)
self.comboBox = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox.setObjectName("comboBox")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.gridLayout.addWidget(self.comboBox, 1, 1, 1, 1)
self.label_59 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_59.setObjectName("label_59")
self.gridLayout.addWidget(self.label_59, 14, 0, 1, 1)
self.comboBox_17 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_17.setObjectName("comboBox_17")
self.comboBox_17.addItem("")
self.comboBox_17.addItem("")
self.comboBox_17.addItem("")
self.comboBox_17.addItem("")
self.comboBox_17.addItem("")
self.gridLayout.addWidget(self.comboBox_17, 9, 1, 1, 1)
self.label_3 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_3.setObjectName("label_3")
self.gridLayout.addWidget(self.label_3, 2, 0, 1, 1)
self.label_12 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_12.setObjectName("label_12")
self.gridLayout.addWidget(self.label_12, 7, 0, 1, 1)
self.label_11 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_11.setObjectName("label_11")
self.gridLayout.addWidget(self.label_11, 6, 0, 1, 1)
self.comboBox_5 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_5.setObjectName("comboBox_5")
self.comboBox_5.addItem("")
self.comboBox_5.addItem("")
self.comboBox_5.addItem("")
self.comboBox_5.addItem("")
self.comboBox_5.addItem("")
self.comboBox_5.addItem("")
self.gridLayout.addWidget(self.comboBox_5, 3, 1, 1, 1)
self.spinBox_5 = QtWidgets.QSpinBox(self.gridLayoutWidget)
```

```
self.spinBox_5.setObjectName("spinBox_5")
self.gridLayout.addWidget(self.spinBox_5, 5, 2, 1, 1)
self.label_9 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_9.setObjectName("label_9")
self.gridLayout.addWidget(self.label_9, 4, 0, 1, 1)
self.spinBox_9 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_9.setObjectName("spinBox_9")
self.gridLayout.addWidget(self.spinBox_9, 9, 2, 1, 1)
self.comboBox_21 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_21.setObjectName("comboBox_21")
self.comboBox_21.addItem("")
self.comboBox_21.addItem("")
self.comboBox_21.addItem("")
self.comboBox_21.addItem("")
self.comboBox_21.addItem("")
self.gridLayout.addWidget(self.comboBox_21, 11, 1, 1, 1)
self.label_13 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_13.setObjectName("label_13")
self.gridLayout.addWidget(self.label_13, 8, 0, 1, 1)
self.comboBox_9 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_9.setObjectName("comboBox_9")
self.comboBox_9.addItem("")
self.comboBox_9.addItem("")
self.comboBox_9.addItem("")
self.comboBox_9.addItem("")
self.comboBox_9.addItem("")
self.gridLayout.addWidget(self.comboBox_9, 5, 1, 1, 1)
self.spinBox_3 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_3.setObjectName("spinBox_3")
self.gridLayout.addWidget(self.spinBox_3, 3, 2, 1, 1)
self.label_4 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_4.setObjectName("label_4")
self.gridLayout.addWidget(self.label_4, 0, 0, 1, 1)
self.label_15 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_15.setObjectName("label_15")
self.gridLayout.addWidget(self.label_15, 10, 0, 1, 1)
self.comboBox_13 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_13.setObjectName("comboBox_13")
self.comboBox_13.addItem("")
self.comboBox_13.addItem("")
self.comboBox_13.addItem("")
self.comboBox_13.addItem("")
self.comboBox_13.addItem("")
self.gridLayout.addWidget(self.comboBox_13, 7, 1, 1, 1)
self.label_16 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_16.setObjectName("label_16")
self.gridLayout.addWidget(self.label_16, 11, 0, 1, 1)
self.spinBox_12 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_12.setObjectName("spinBox_12")
self.gridLayout.addWidget(self.spinBox_12, 12, 2, 1, 1)
self.label_2 = QtWidgets.QLabel(self.gridLayoutWidget)
```

```
self.label_2.setObjectName("label_2")
self.gridLayout.addWidget(self.label_2, 1, 0, 1, 1)
self.spinBox = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox.setObjectName("spinBox")
self.gridLayout.addWidget(self.spinBox, 1, 2, 1, 1)
self.label_5 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_5.setObjectName("label_5")
self.gridLayout.addWidget(self.label_5, 0, 1, 1, 1)
self.comboBox_11 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_11.setObjectName("comboBox_11")
self.comboBox_11.addItem("")
self.comboBox_11.addItem("")
self.comboBox_11.addItem("")
self.comboBox_11.addItem("")
self.comboBox_11.addItem("")
self.gridLayout.addWidget(self.comboBox_11, 6, 1, 1, 1)
self.comboBox_7 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_7.setObjectName("comboBox_7")
self.comboBox_7.addItem("")
self.comboBox_7.addItem("")
self.comboBox_7.addItem("")
self.comboBox_7.addItem("")
self.comboBox_7.addItem("")
self.gridLayout.addWidget(self.comboBox_7, 4, 1, 1, 1)
self.label_58 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_58.setObjectName("label_58")
self.gridLayout.addWidget(self.label_58, 13, 0, 1, 1)
self.spinBox_6 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_6.setObjectName("spinBox_6")
self.gridLayout.addWidget(self.spinBox_6, 6, 2, 1, 1)
self.label_17 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_17.setObjectName("label_17")
self.gridLayout.addWidget(self.label_17, 12, 0, 1, 1)
self.spinBox_8 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_8.setObjectName("spinBox_8")
self.gridLayout.addWidget(self.spinBox_8, 8, 2, 1, 1)
self.label_8 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_8.setObjectName("label_8")
self.gridLayout.addWidget(self.label_8, 3, 0, 1, 1)
self.spinBox_11 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_11.setObjectName("spinBox_11")
self.gridLayout.addWidget(self.spinBox_11, 11, 2, 1, 1)
self.comboBox_23 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_23.setObjectName("comboBox_23")
self.comboBox_23.addItem("")
self.comboBox_23.addItem("")
self.comboBox_23.addItem("")
self.comboBox_23.addItem("")
self.comboBox_23.addItem("")
self.gridLayout.addWidget(self.comboBox_23, 12, 1, 1, 1)
self.spinBox_10 = QtWidgets.QSpinBox(self.gridLayoutWidget)
```

```
self.spinBox_10.setObjectName("spinBox_10")
self.gridLayout.addWidget(self.spinBox_10, 10, 2, 1, 1)
self.spinBox_2 = QtWidgets.QSpinBox(self.gridLayoutWidget)
self.spinBox_2.setObjectName("spinBox_2")
self.gridLayout.addWidget(self.spinBox_2, 2, 2, 1, 1)
self.label_7 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_7.setObjectName("label_7")
self.gridLayout.addWidget(self.label_7, 0, 2, 1, 1)
self.comboBox_15 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_15.setObjectName("comboBox_15")
self.comboBox_15.addItem("")
self.comboBox_15.addItem("")
self.comboBox_15.addItem("")
self.comboBox_15.addItem("")
self.comboBox_15.addItem("")
self.gridLayout.addWidget(self.comboBox_15, 8, 1, 1, 1)
self.comboBox_19 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_19.setObjectName("comboBox_19")
self.comboBox_19.addItem("")
self.comboBox_19.addItem("")
self.comboBox_19.addItem("")
self.comboBox_19.addItem("")
self.comboBox_19.addItem("")
self.gridLayout.addWidget(self.comboBox_19, 10, 1, 1, 1)
self.label_14 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_14.setObjectName("label_14")
self.gridLayout.addWidget(self.label_14, 9, 0, 1, 1)
self.label_60 = QtWidgets.QLabel(self.gridLayoutWidget)
self.label_60.setObjectName("label_60")
self.gridLayout.addWidget(self.label_60, 15, 0, 1, 1)
self.comboBox_32 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_32.setObjectName("comboBox_32")
self.comboBox_32.addItem("")
self.comboBox_32.addItem("")
self.comboBox_32.addItem("")
self.comboBox_32.addItem("")
self.comboBox_32.addItem("")
self.gridLayout.addWidget(self.comboBox_32, 13, 1, 1, 1)
self.comboBox_34 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_34.setObjectName("comboBox_34")
self.comboBox_34.addItem("")
self.comboBox_34.addItem("")
self.comboBox_34.addItem("")
self.comboBox_34.addItem("")
self.comboBox_34.addItem("")
self.gridLayout.addWidget(self.comboBox_34, 14, 1, 1, 1)
self.comboBox_36 = QtWidgets.QComboBox(self.gridLayoutWidget)
self.comboBox_36.setObjectName("comboBox_36")
self.comboBox_36.addItem("")
self.comboBox_36.addItem("")
self.comboBox_36.addItem("")
```

```
        self.comboBox_36.addItem("")
        self.comboBox_36.addItem("")
        self.gridLayout.addWidget(self.comboBox_36, 15, 1, 1, 1)
        self.spinBox_40 = QtWidgets.QSpinBox(self.gridLayoutWidget)
        self.spinBox_40.setObjectName("spinBox_40")
        self.gridLayout.addWidget(self.spinBox_40, 13, 2, 1, 1)
        self.spinBox_41 = QtWidgets.QSpinBox(self.gridLayoutWidget)
        self.spinBox_41.setObjectName("spinBox_41")
        self.gridLayout.addWidget(self.spinBox_41, 14, 2, 1, 1)
        self.spinBox_42 = QtWidgets.QSpinBox(self.gridLayoutWidget)
        self.spinBox_42.setObjectName("spinBox_42")
        self.gridLayout.addWidget(self.spinBox_42,  15,  2,  1,  1)
        self.label = QtWidgets.QLabel(self.sgpa_with_grades)
        self.label.setGeometry(QtCore.QRect(10, 0, 621, 41))
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.label_18 = QtWidgets.QLabel(self.sgpa_with_grades)
        self.label_18.setGeometry(QtCore.QRect(50, 580, 361, 41))
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        self.label_18.setFont(font)
        self.label_18.setObjectName("label_18")
        self.lcdNumber = QtWidgets.QLCDNumber(self.sgpa_with_grades)
        self.lcdNumber.setGeometry(QtCore.QRect(450, 580, 131, 41))
        self.lcdNumber.setStyleSheet("color: rgb(0, 255, 0);\n"
"background-color: rgb(0, 0, 0);")
        self.lcdNumber.setObjectName("lcdNumber")
        self.label_19 = QtWidgets.QLabel(self.sgpa_with_grades)
        self.label_19.setGeometry(QtCore.QRect(110, 720, 571, 61))
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        self.label_19.setFont(font)
        self.label_19.setObjectName("label_19")
        self.label_20 = QtWidgets.QLabel(self.sgpa_with_grades)
        self.label_20.setGeometry(QtCore.QRect(20, 800, 281, 51))
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        self.label_20.setFont(font)
        self.label_20.setObjectName("label_20")
        self.pushButton = QtWidgets.QPushButton(self.sgpa_with_grades)
        self.pushButton.setGeometry(QtCore.QRect(270, 500, 131, 31))
```

```
        self.gridLayout_4.addWidget(self.label_77, 1, 2, 1, 1)
        self.label_72 = QtWidgets.QLabel(self.gridLayoutWidget_4)
        self.label_72.setObjectName("label_72")
        self.gridLayout_4.addWidget(self.label_72, 4, 0, 1, 1)
        self.label_74 = QtWidgets.QLabel(self.gridLayoutWidget_4)
        self.label_74.setObjectName("label_74")
        self.gridLayout_4.addWidget(self.label_74, 6, 0, 1, 1)
        self.label_83 = QtWidgets.QLabel(self.gridLayoutWidget_4)
        self.label_83.setObjectName("label_83")
        self.gridLayout_4.addWidget(self.label_83, 7, 2, 1, 1)
        self.spinBox_51 = QtWidgets.QSpinBox(self.gridLayoutWidget_4)
        self.spinBox_51.setMaximum(500)
        self.spinBox_51.setObjectName("spinBox_51")
        self.gridLayout_4.addWidget(self.spinBox_51, 5, 1, 1, 1)
        self.spinBox_53 = QtWidgets.QSpinBox(self.gridLayoutWidget_4)
        self.spinBox_53.setMaximum(500)


    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "SGPA & CGPA
CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N"))
        self.comboBox_3.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_3.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_3.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_3.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_3.setItemText(4,  _translate("MainWindow",  "4"))
        self.label_10.setText(_translate("MainWindow", "05"))
        self.comboBox.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox.setItemText(4,  _translate("MainWindow",  "4"))
        self.label_59.setText(_translate("MainWindow", "14"))
        self.comboBox_17.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_17.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_17.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_17.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_17.setItemText(4,  _translate("MainWindow",  "4"))
        self.label_3.setText(_translate("MainWindow", "02"))
        self.label_12.setText(_translate("MainWindow", "07"))
        self.label_11.setText(_translate("MainWindow", "06"))
        self.comboBox_5.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_5.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_5.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_5.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_5.setItemText(4,  _translate("MainWindow",  "4"))
        self.comboBox_5.setItemText(5,  _translate("MainWindow",  "8"))
        self.label_9.setText(_translate("MainWindow", "04"))
        self.comboBox_21.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_21.setItemText(1,  _translate("MainWindow",  "1"))
```

```
        self.comboBox_36.setItemText(4, _translate("MainWindow", "4"))
        self.label.setText(_translate("MainWindow", "CALCULATE SGPA WITH GRADES:-"))
        self.label_18.setText(_translate("MainWindow", "SEMESTER GRADE POINT
AVERAGE:-"))
        self.label_19.setText(_translate("MainWindow", "DESIGNED AND DEVELOPED BY
ADARSH N"))
        self.label_20.setText(_translate("MainWindow", "FOR VTU STUDENTS ONLY "))
        self.pushButton.setText(_translate("MainWindow", "CLEAR"))
        self.pushButton_2.setText(_translate("MainWindow", "CALCULATE"))
        self.pushButton_5.setText(_translate("MainWindow", "USER MANUL/SOUCE CODE"))
        self.pushButton_40.setText(_translate("MainWindow", "5th sem"))
        self.pushButton_41.setText(_translate("MainWindow", "8th sem"))
        self.pushButton_42.setText(_translate("MainWindow", "3rd sem"))
        self.pushButton_43.setText(_translate("MainWindow", "6th sem"))
        self.pushButton_44.setText(_translate("MainWindow", "4th sem"))
        self.pushButton_45.setText(_translate("MainWindow", "1st sem"))
        self.pushButton_46.setText(_translate("MainWindow", "7th sem"))
        self.pushButton_47.setText(_translate("MainWindow", "2nd sem"))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.sgpa_with_grades),
_translate("MainWindow", "SGPA WITH GRADES"))
        self.label_21.setText(_translate("MainWindow", "CALCULATE SGPA WITH MARKS:-
"))
        self.label_25.setText(_translate("MainWindow", "Total Marks of Subject:-"))
        self.comboBox_25.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_25.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_25.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_25.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_25.setItemText(4, _translate("MainWindow", "4"))
        self.label_23.setText(_translate("MainWindow", "Credits Assigned:-"))
        self.label_37.setText(_translate("MainWindow", "12"))
        self.label_26.setText(_translate("MainWindow", "01"))
        self.label_61.setText(_translate("MainWindow", "13"))
        self.label_32.setText(_translate("MainWindow", "07"))
        self.comboBox_29.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_29.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_29.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_29.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_29.setItemText(4,  _translate("MainWindow",  "4"))
        self.comboBox_29.setItemText(5,  _translate("MainWindow",  "8"))
        self.comboBox_33.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_33.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_33.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_33.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_33.setItemText(4,  _translate("MainWindow",  "4"))
        self.comboBox_43.setItemText(0,  _translate("MainWindow",  "0"))
        self.comboBox_43.setItemText(1,  _translate("MainWindow",  "1"))
        self.comboBox_43.setItemText(2,  _translate("MainWindow",  "2"))
        self.comboBox_43.setItemText(3,  _translate("MainWindow",  "3"))
        self.comboBox_43.setItemText(4,  _translate("MainWindow",  "4"))
        self.label_30.setText(_translate("MainWindow", "05"))
        self.label_34.setText(_translate("MainWindow", "09"))
```

```
        self.comboBox_35.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_35.setItemText(1, _translate("MainWindow", "1"))
        self.comboBox_35.setItemText(2, _translate("MainWindow", "2"))
        self.comboBox_35.setItemText(3, _translate("MainWindow", "3"))
        self.comboBox_35.setItemText(4, _translate("MainWindow", "4"))
        self.label_27.setText(_translate("MainWindow", "02"))
        self.comboBox_27.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_27.setItemText(1, _translate("MainWindow", "1"))
        self.comboBox_27.setItemText(2, _translate("MainWindow", "2"))
        self.comboBox_27.setItemText(3, _translate("MainWindow", "3"))
        self.comboBox_27.setItemText(4, _translate("MainWindow", "4"))
        self.label_22.setText(_translate("MainWindow", "Subject No:-"))
        self.comboBox_37.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_37.setItemText(1, _translate("MainWindow", "1"))
        self.comboBox_37.setItemText(2, _translate("MainWindow", "2"))
        self.comboBox_37.setItemText(3, _translate("MainWindow", "3"))
        self.comboBox_37.setItemText(4, _translate("MainWindow", "4"))
        self.label_29.setText(_translate("MainWindow", "04"))
        self.label_31.setText(_translate("MainWindow", "06"))
        self.label_33.setText(_translate("MainWindow", "08"))
        self.comboBox_47.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_47.setItemText(1, _translate("MainWindow", "1"))
        self.comboBox_47.setItemText(2, _translate("MainWindow", "2"))
        self.comboBox_47.setItemText(3, _translate("MainWindow", "3"))
        self.comboBox_47.setItemText(4, _translate("MainWindow", "4"))
        self.label_35.setText(_translate("MainWindow", "10"))
        self.comboBox_45.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_45.setItemText(1, _translate("MainWindow", "1"))
        self.comboBox_45.setItemText(2, _translate("MainWindow", "2"))
        self.comboBox_45.setItemText(3, _translate("MainWindow", "3"))
        self.comboBox_45.setItemText(4, _translate("MainWindow", "4"))
        self.label_36.setText(_translate("MainWindow", "11"))
        self.comboBox_31.setItemText(0, _translate("MainWindow", "0"))
        self.comboBox_31.setItemText(1, _translate("MainWindow", "1"))
        self.comboBox_31.setItemText(2, _translate("MainWindow", "2"))
        self.comboBox_31.setItemText(3, _translate("MainWindow", "3"))
        self.comboBox_31.setItemText(4, _translate("MainWindow", "4"))
        self.label_28.setText(_translate("MainWindow", "03"))
CALCULATOR PART - 02"))
        self.pushButton_2.clicked.connect(self. calculate_1)
        self.pushButton_4.clicked.connect(self. calculate_2)
        self.pushButton_14.clicked.connect(self.calculate_3)
        self.pushButton_8.clicked.connect(self.calculate_4)
        self.pushButton_12.clicked.connect(self.calculate_5)
        self.pushButton_23.clicked.connect(self. cgpa_two_eight)
        self.pushButton_45.clicked.connect(self. sgpa_one_one_and_two)

    def openlink(self):
            webbrowser.open('https://github.com/Adarsh232001/VTU-SGPA-CALCULATOR')

    def one_and_two(self):
```

```
        self.comboBox_2.setCurrentIndex(4)
        self.comboBox_4.setCurrentIndex(4)
        self.comboBox_6.setCurrentIndex(3)
        self.comboBox_8.setCurrentIndex(3)
        self.comboBox_10.setCurrentIndex(3)
        self.comboBox_12.setCurrentIndex(1)
        self.comboBox_14.setCurrentIndex(1)
        self.comboBox_16.setCurrentIndex(1)
        clear =
['comboBox_18','comboBox_20','comboBox_22','comboBox_24','comboBox_26','comboBox_28'
,'comboBox_30']
        clear_comboBox(self, clear)

    def one_and_two_(self):
        self.comboBox_46.setCurrentIndex(4)
        self.comboBox_44.setCurrentIndex(4)
        self.comboBox_49.setCurrentIndex(3)
        self.comboBox_54.setCurrentIndex(3)
        self.comboBox_51.setCurrentIndex(3)
        self.comboBox_53.setCurrentIndex(1)
        self.comboBox_52.setCurrentIndex(1)
        self.comboBox_56.setCurrentIndex(1)
        clear =
['comboBox_48','comboBox_57','comboBox_50','comboBox_55','comboBox_58','comboBox_59'
,'comboBox_60']
        clear_comboBox(self, clear)

    def three_and_four(self):
        self.comboBox_2.setCurrentIndex(3)
        self.comboBox_4.setCurrentIndex(4)
        self.comboBox_6.setCurrentIndex(3)
        self.comboBox_8.setCurrentIndex(3)
        self.comboBox_10.setCurrentIndex(3)
        self.comboBox_12.setCurrentIndex(3)
        self.comboBox_14.setCurrentIndex(2)
        self.comboBox_16.setCurrentIndex(2)
        self.comboBox_18.setCurrentIndex(1)
        clear =self.comboBox_10.setCurrentIndex(3)



    def sgpa_two_seven(self):
        self.comboBox_25.setCurrentIndex(4)
        self.comboBox_27.setCurrentIndex(4)
        self.comboBox_29.setCurrentIndex(3)
        self.comboBox_31.setCurrentIndex(3)
        self.comboBox_33.setCurrentIndex(3)
        self.comboBox_35.setCurrentIndex(2)
        self.comboBox_37.setCurrentIndex(1)
        clear =
['comboBox_39','comboBox_41','comboBox_43','comboBox_45','comboBox_47','comboBox_38'
```

```
,'comboBox_40','comboBox_42']
        clear_comboBox(self,  clear)

    def sgpa_two_eight(self):
        self.comboBox_25.setCurrentIndex(3)
        self.comboBox_27.setCurrentIndex(3)
        self.comboBox_29.setCurrentIndex(5)
        self.comboBox_31.setCurrentIndex(1)
        self.comboBox_33.setCurrentIndex(3)
        clear =
['comboBox_35','comboBox_37','comboBox_39','comboBox_41','comboBox_43','comboBox_45'
,'comboBox_47','comboBox_38','comboBox_40','comboBox_42']
        clear_comboBox(self, clear)

    def clear_1(self):
        self.comboBox.setCurrentIndex(0)
        self.comboBox_3.setCurrentIndex(0)
        self.comboBox_5.setCurrentIndex(0)
        self.comboBox_7.setCurrentIndex(0)
        self.comboBox_9.setCurrentIndex(0)
        self.comboBox_11.setCurrentIndex(0)
        self.comboBox_13.setCurrentIndex(0)
        self.comboBox_15.setCurrentIndex(0)
        self.comboBox_17.setCurrentIndex(0)
        self.comboBox_19.setCurrentIndex(0)
        self.comboBox_21.setCurrentIndex(0)
        self.comboBox_23.setCurrentIndex(0)
        self.comboBox_32.setCurrentIndex(0)
        self.comboBox_34.setCurrentIndex(0)
        self.comboBox_36.setCurrentIndex(0)
        self.spinBox.setValue(0)
        self.spinBox_2.setValue(0)
        self.spinBox_3.setValue(0)
        self.spinBox_4.setValue(0)
        self.spinBox_5.setValue(0)
        self.spinBox_6.setValue(0)
        self.spinBox_7.setValue(0)
        self.spinBox_8.setValue(0)
        self.spinBox_9.setValue(0)
        self.spinBox_10.setValue(0)
        self.spinBox_11.setValue(0)
        self.spinBox_12.setValue(0)
        self.spinBox_40.setValue(0)
        self.spinBox_41.setValue(0)
        self.spinBox_42.setValue(0)
        grade_points.clear()
        credits.clear()
        total_marks.clear()
        self.lcdNumber.display(0)

    #reset the tab
```

```
    def clear_2(self):
        self.spinBox_13.setValue(0)
        self.spinBox_14.setValue(0)
        self.spinBox_15.setValue(0)
        self.spinBox_16.setValue(0)
        self.spinBox_17.setValue(0)
        self.spinBox_18.setValue(0)
        self.spinBox_19.setValue(0)
        self.spinBox_20.setValue(0)
        self.spinBox_21.setValue(0)
        self.spinBox_22.setValue(0)
        self.spinBox_23.setValue(0)
        self.spinBox_24.setValue(0)
        self.spinBox_43.setValue(0)
        self.spinBox_44.setValue(0)
        self.spinBox_45.setValue(0)
        self.comboBox_25.setCurrentIndex(0)
        self.comboBox_27.setCurrentIndex(0)
        self.comboBox_29.setCurrentIndex(0)
        self.comboBox_31.setCurrentIndex(0)
        self.comboBox_33.setCurrentIndex(0)
        self.comboBox_35.setCurrentIndex(0)
        self.comboBox_37.setCurrentIndex(0)
        self.comboBox_39.setCurrentIndex(0)
        self.comboBox_41.setCurrentIndex(0)
        self.comboBox_43.setCurrentIndex(0)
        self.comboBox_45.setCurrentIndex(0)
        self.comboBox_47.setCurrentIndex(0)
        self.comboBox_38.setCurrentIndex(0)
        self.comboBox_40.setCurrentIndex(0)
        self.comboBox_42.setCurrentIndex(0)
        grade_points.clear()
        credits.clear()
        total_marks.clear()
        self.lcdNumber_2.display(0)

    def clear_3(self):
        self.comboBox_46.setCurrentIndex(0)
        self.comboBox_44.setCurrentIndex(0)
        self.comboBox_49.setCurrentIndex(0)
        self.comboBox_54.setCurrentIndex(0)
        self.comboBox_51.setCurrentIndex(0)
        self.comboBox_53.setCurrentIndex(0)
        self.comboBox_52.setCurrentIndex(0)
        self.comboBox_56.setCurrentIndex(0)
        self.comboBox_48.setCurrentIndex(0)
        self.comboBox_57.setCurrentIndex(0)
        self.comboBox_50.setCurrentIndex(0)
        self.comboBox_55.setCurrentIndex(0)
        self.comboBox_58.setCurrentIndex(0)
        self.comboBox_59.setCurrentIndex(0)
```

```
        self.comboBox_60.setCurrentIndex(0)
        self.spinBox_68.setValue(0)
        self.spinBox_73.setValue(0)
        self.spinBox_66.setValue(0)
        self.spinBox_63.setValue(0)
        self.spinBox_64.setValue(0)
        self.spinBox_69.setValue(0)
        self.spinBox_62.setValue(0)
        self.spinBox_70.setValue(0)
        self.spinBox_65.setValue(0)
        self.spinBox_72.setValue(0)
        self.spinBox_71.setValue(0)
        self.spinBox_67.setValue(0)
        self.spinBox_74.setValue(0)
        self.spinBox_75.setValue(0)
        self.spinBox_76.setValue(0)
        grade_points.clear()
        credits.clear()
        credits_earned.clear()
        total_marks.clear()
        self.lcdNumber_6.display(0)
        self.lcdNumber_7.display(0)

    def clear_4(self):
        self.spinBox_25.setValue(0)
        self.spinBox_26.setValue(0)
        self.spinBox_27.setValue(0)
        self.spinBox_28.setValue(0)
        self.spinBox_29.setValue(0)
        self.spinBox_30.setValue(0)
        self.spinBox_31.setValue(0)
        self.spinBox_32.setValue(0)
        self.spinBox_33.setValue(0)
        self.spinBox_34.setValue(0)
        self.spinBox_35.setValue(0)
        self.spinBox_36.setValue(0)
        self.spinBox_37.setValue(0)
        self.spinBox_38.setValue(0)
        self.spinBox_39.setValue(0)
        self.comboBox_2.setCurrentIndex(0)
        self.comboBox_4.setCurrentIndex(0)
        self.comboBox_6.setCurrentIndex(0)
        self.comboBox_8.setCurrentIndex(0)
        self.comboBox_10.setCurrentIndex(0)
        self.comboBox_12.setCurrentIndex(0)
        self.comboBox_14.setCurrentIndex(0)
        self.comboBox_16.setCurrentIndex(0)
        self.comboBox_18.setCurrentIndex(0)
        self.comboBox_20.setCurrentIndex(0)
        self.comboBox_22.setCurrentIndex(0)
        self.comboBox_24.setCurrentIndex(0)
```

```
        self.comboBox_26.setCurrentIndex(0)
        self.comboBox_28.setCurrentIndex(0)
        self.comboBox_30.setCurrentIndex(0)
        grade_points.clear()
        credits.clear()
        credits_earned.clear()
        total_marks.clear()
        self.lcdNumber_3.display(0)
        self.lcdNumber_4.display(0)

    def clear_5(self):
        self.spinBox_46.setValue(0)
        self.spinBox_47.setValue(0)
        self.spinBox_48.setValue(0)
        self.spinBox_49.setValue(0)
        self.spinBox_50.setValue(0)
        self.spinBox_51.setValue(0)
        self.spinBox_52.setValue(0)
        self.spinBox_53.setValue(0)
        self.spinBox_54.setValue(0)
        self.spinBox_55.setValue(0)
        self.spinBox_56.setValue(0)
        self.spinBox_57.setValue(0)
        self.spinBox_58.setValue(0)
        self.spinBox_59.setValue(0)
        self.spinBox_60.setValue(0)
        self.spinBox_61.setValue(0)
        self.lcdNumber_5.display(0)
        cixgi.clear()
        ci_earned.clear()

    def calculate_1(self):
            credits_0 = int(self.comboBox.currentText())
            credits.append(credits_0)
            credits_1 = int(self.comboBox_3.currentText())
            credits.append(credits_1)
            credits_2 = int(self.comboBox_5.currentText())
            credits.append(credits_2)
            credits_3 = int(self.comboBox_7.currentText())
            credits.append(credits_3)
            credits_4 = int(self.comboBox_9.currentText())
            credits.append(credits_4)
            credits_5 = int(self.comboBox_11.currentText())
            credits.append(credits_5)
            credits_6 = int(self.comboBox_13.currentText())
            credits.append(credits_6)
            credits_7 = int(self.comboBox_15.currentText())
            credits.append(credits_7)
            credits_8 = int(self.comboBox_17.currentText())
            credits.append(credits_8)
            credits_9 = int(self.comboBox_19.currentText())
```

```
credits.append(credits_9)
credits_10 = int(self.comboBox_21.currentText())
credits.append(credits_10)
credits_11 = int(self.comboBox_23.currentText())
credits.append(credits_11)
credits_12 = int(self.comboBox_32.currentText())
credits.append(credits_12)
credits_13 = int(self.comboBox_34.currentText())
credits.append(credits_13)
credits_14 = int(self.comboBox_36.currentText())
credits.append(credits_14)
grade_0 = self.spinBox.value()
check_grade(self, grade_0)
grade_points.append(grade_0)
grade_1 = self.spinBox_2.value()
check_grade(self, grade_1)
grade_points.append(grade_1)
grade_2 = self.spinBox_3.value()
check_grade(self, grade_2)
grade_points.append(grade_2)
grade_3 = self.spinBox_4.value()
check_grade(self, grade_3)
grade_points.append(grade_3)
grade_4 = self.spinBox_5.value()
check_grade(self, grade_4)
grade_points.append(grade_4)
grade_5 = self.spinBox_6.value()
check_grade(self, grade_5)
grade_points.append(grade_5)
grade_6 = self.spinBox_7.value()
check_grade(self, grade_6)
grade_points.append(grade_6)
grade_7 = self.spinBox_8.value()
check_grade(self, grade_7)
grade_points.append(grade_7)
grade_8 = self.spinBox_9.value()
check_grade(self, grade_8)
grade_points.append(grade_8)
grade_9 = self.spinBox_10.value()
check_grade(self, grade_9)
grade_points.append(grade_9)
grade_10 = self.spinBox_11.value()
check_grade(self, grade_10)
grade_points.append(grade_10)
grade_11 = self.spinBox_12.value()
check_grade(self, grade_11)
grade_points.append(grade_11)
grade_12 = self.spinBox_40.value()
check_grade(self, grade_12)
grade_points.append(grade_12)
grade_13 = self.spinBox_41.value()
```

```
        check_grade(self, grade_13)
        grade_points.append(grade_13)
        grade_14 = self.spinBox_42.value()
        check_grade(self, grade_14)
        grade_points.append(grade_14)
        calculate_with_grades(self, lcdNumber)

    def calculate_3(self):
        credits_0 = int(self.comboBox_46.currentText())
        credits.append(credits_0)
        credits_1 = int(self.comboBox_44.currentText())
        credits.append(credits_1)
        credits_2 = int(self.comboBox_49.currentText())
        credits.append(credits_2)
        credits_3 = int(self.comboBox_54.currentText())
        credits.append(credits_3)
        credits_4 = int(self.comboBox_51.currentText())
        credits.append(credits_4)
        credits_5 = int(self.comboBox_53.currentText())
        credits.append(credits_5)
        credits_6 = int(self.comboBox_52.currentText())
        credits.append(credits_6)
        credits_7 = int(self.comboBox_56.currentText())
        credits.append(credits_7)
        credits_8 = int(self.comboBox_48.currentText())
        credits.append(credits_8)
        credits_9 = int(self.comboBox_57.currentText())
        credits.append(credits_9)
        credits_10 = int(self.comboBox_50.currentText())
        credits.append(credits_10)
        credits_11 = int(self.comboBox_55.currentText())
        credits.append(credits_11)
        credits_12 = int(self.comboBox_58.currentText())
        credits.append(credits_12)
        credits_13 = int(self.comboBox_59.currentText())
        credits.append(credits_13)
        credits_14 = int(self.comboBox_60.currentText())
        credits.append(credits_14)
        grade_0 = self.spinBox_68.value()
        check_grade(self, grade_0)
        grade_points.append(grade_0)
        grade_1 = self.spinBox_73.value()
        check_grade(self, grade_1)
        grade_points.append(grade_1)
        grade_2 = self.spinBox_66.value()
        check_grade(self, grade_2)
        grade_points.append(grade_2)
        grade_3 = self.spinBox_63.value()
        check_grade(self, grade_3)
        grade_points.append(grade_3)
        grade_4 = self.spinBox_64.value()
```

```
        check_grade(self, grade_4)
        grade_points.append(grade_4)
        grade_5 = self.spinBox_69.value()
        check_grade(self, grade_5)
        grade_points.append(grade_5)
        grade_6 = self.spinBox_62.value()
        check_grade(self, grade_6)
        grade_points.append(grade_6)
        grade_7 = self.spinBox_70.value()
        check_grade(self, grade_7)
        grade_points.append(grade_7)
        grade_8 = self.spinBox_65.value()
        check_grade(self, grade_8)
        grade_points.append(grade_8)
        grade_9 = self.spinBox_72.value()
        check_grade(self, grade_9)
        grade_points.append(grade_9)
        grade_10 = self.spinBox_71.value()
        check_grade(self, grade_10)
        grade_points.append(grade_10)
        grade_11 = self.spinBox_67.value()
        check_grade(self, grade_11)
        grade_points.append(grade_11)
        grade_12 = self.spinBox_74.value()
        check_grade(self, grade_12)
        grade_points.append(grade_12)
        grade_13 = self.spinBox_75.value()
        check_grade(self, grade_13)
        grade_points.append(grade_13)
        grade_14 = self.spinBox_76.value()
        check_grade(self, grade_14)
        grade_points.append(grade_14)
        for i in range(len(grade_points)):
            if (credits[i] * grade_points[i]) > 0:
                credits_earned.append(credits[i])
        calculate_cgpa_with_grades(self,  lcdNumber_1)

    def calculate_4(self):
        lcdNumber_1 = False
        credits_0 = int(self.comboBox_2.currentText())
        credits.append(credits_0)
        credits_1 = int(self.comboBox_4.currentText())
        credits.append(credits_1)
        credits_2 = int(self.comboBox_6.currentText())
        credits.append(credits_2)
        credits_3 = int(self.comboBox_8.currentText())
        credits.append(credits_3)
        credits_4 = int(self.comboBox_10.currentText())
        credits.append(credits_4)
        credits_5 = int(self.comboBox_12.currentText())
        credits.append(credits_5)
```

```
credits_6 = int(self.comboBox_14.currentText())
credits.append(credits_6)
credits_7 = int(self.comboBox_16.currentText())
credits.append(credits_7)
credits_8 = int(self.comboBox_18.currentText())
credits.append(credits_8)
credits_9 = int(self.comboBox_20.currentText())
credits.append(credits_9)
credits_10 = int(self.comboBox_22.currentText())
credits.append(credits_10)
credits_11 = int(self.comboBox_24.currentText())
credits.append(credits_11)
credits_12 = int(self.comboBox_26.currentText())
credits.append(credits_12)
credits_13 = int(self.comboBox_28.currentText())
credits.append(credits_13)
credits_14 = int(self.comboBox_30.currentText())
credits.append(credits_14)
marks_0 = self.spinBox_25.value()
total_marks.append(marks_0)
marks_1 = self.spinBox_26.value()
total_marks.append(marks_1)
marks_2 = self.spinBox_27.value()
total_marks.append(marks_2)
marks_3 = self.spinBox_28.value()
total_marks.append(marks_3)
marks_4 = self.spinBox_29.value()
total_marks.append(marks_4)
marks_5 = self.spinBox_30.value()
total_marks.append(marks_5)
marks_6 = self.spinBox_31.value()
total_marks.append(marks_6)
marks_7 = self.spinBox_32.value()
total_marks.append(marks_7)
marks_8 = self.spinBox_33.value()
total_marks.append(marks_8)
marks_9 = self.spinBox_34.value()
total_marks.append(marks_9)
marks_10 = self.spinBox_35.value()
total_marks.append(marks_10)
marks_11 = self.spinBox_36.value()
total_marks.append(marks_11)
marks_12 = self.spinBox_37.value()
total_marks.append(marks_12)
marks_13 = self.spinBox_38.value()
total_marks.append(marks_13)
marks_14 = self.spinBox_39.value()
total_marks.append(marks_14)
marks_to_grades()
credits_earned.clear()
for i in range(len(grade_points)):
```

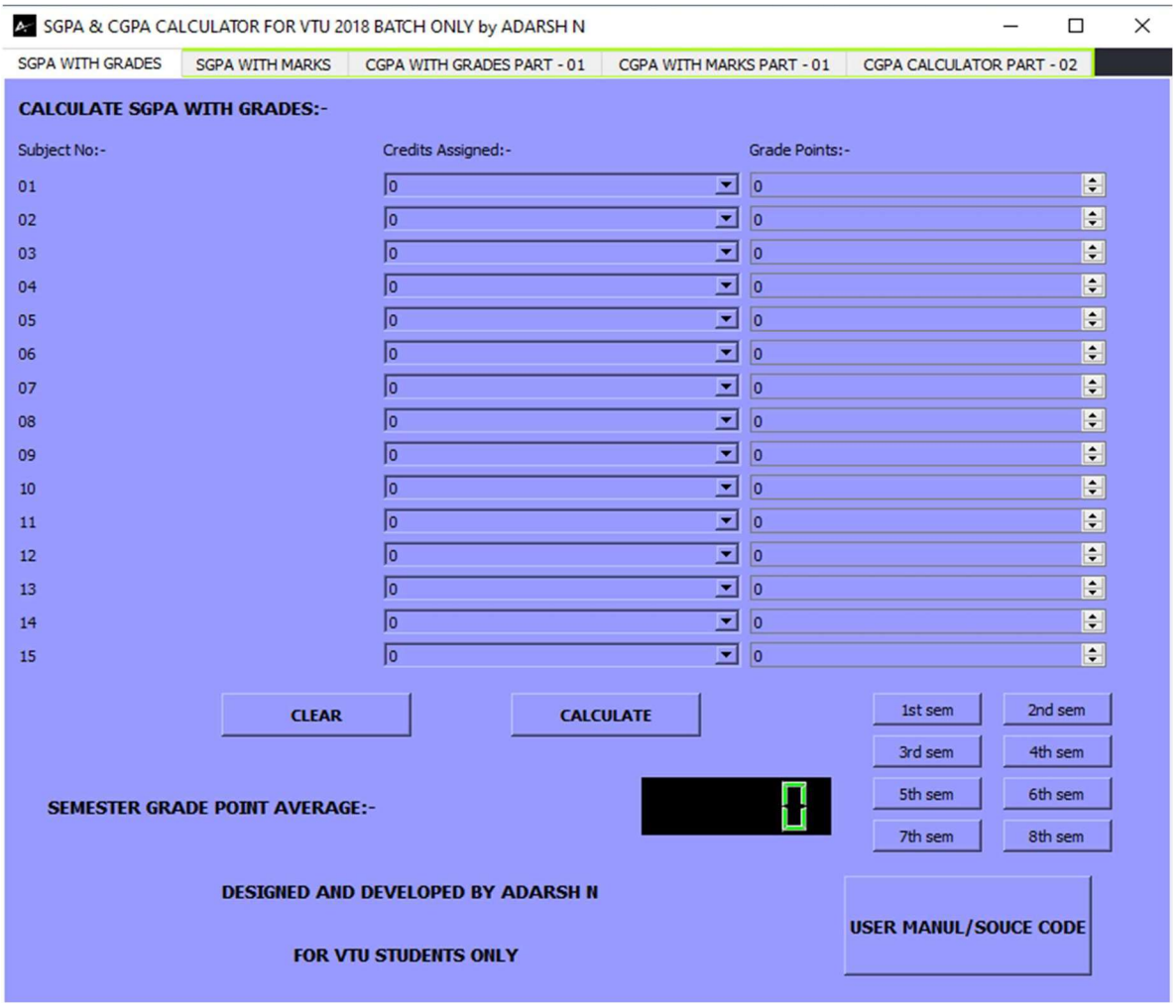
```
            if (credits[i] * grade_points[i]) > 0:
                credits_earned.append(credits[i])
        calculate_cgpa_with_grades(self,  lcdNumber_1)

    def calculate_5(self):
        ci_gi = self.spinBox_46.value()
        cixgi.append(ci_gi)
        ci_gi_1 = self.spinBox_47.value()
        cixgi.append(ci_gi_1)
        ci_gi_2 = self.spinBox_48.value()
        cixgi.append(ci_gi_2)
        ci_gi_3 = self.spinBox_49.value()
        cixgi.append(ci_gi_3)
        ci_gi_4 = self.spinBox_50.value()
        cixgi.append(ci_gi_4)
        ci_gi_5 = self.spinBox_51.value()
        cixgi.append(ci_gi_5)
        ci_gi_6 = self.spinBox_52.value()
        cixgi.append(ci_gi_6)
        ci_gi_7 = self.spinBox_53.value()
        cixgi.append(ci_gi_7)
        ci_e = self.spinBox_54.value()
        ci_earned.append(ci_e)
        ci_e_1 = self.spinBox_55.value()
        ci_earned.append(ci_e_1)
        ci_e_2 = self.spinBox_56.value()
        ci_earned.append(ci_e_2)
        ci_e_3 = self.spinBox_57.value()
        ci_earned.append(ci_e_3)
        ci_e_4 = self.spinBox_58.value()
        ci_earned.append(ci_e_4)
        ci_e_5 = self.spinBox_59.value()
        ci_earned.append(ci_e_5)
        ci_e_6 = self.spinBox_60.value()
        ci_earned.append(ci_e_6)
        ci_e_7 = self.spinBox_61.value()
        ci_earned.append(ci_e_7)
        calculate_cgpa(self)

def clear_comboBox(self, clear):
    for i in clear:
        eval('self.'+i+'.setCurrentIndex(0)')

def calculate_cgpa(self):
    try:

        final_cgpa = sum(cixgi) / sum(ci_earned)
        self.lcdNumber_5.display(final_cgpa)

    except ZeroDivisionError:
        msg = QtWidgets.QMessageBox()
```

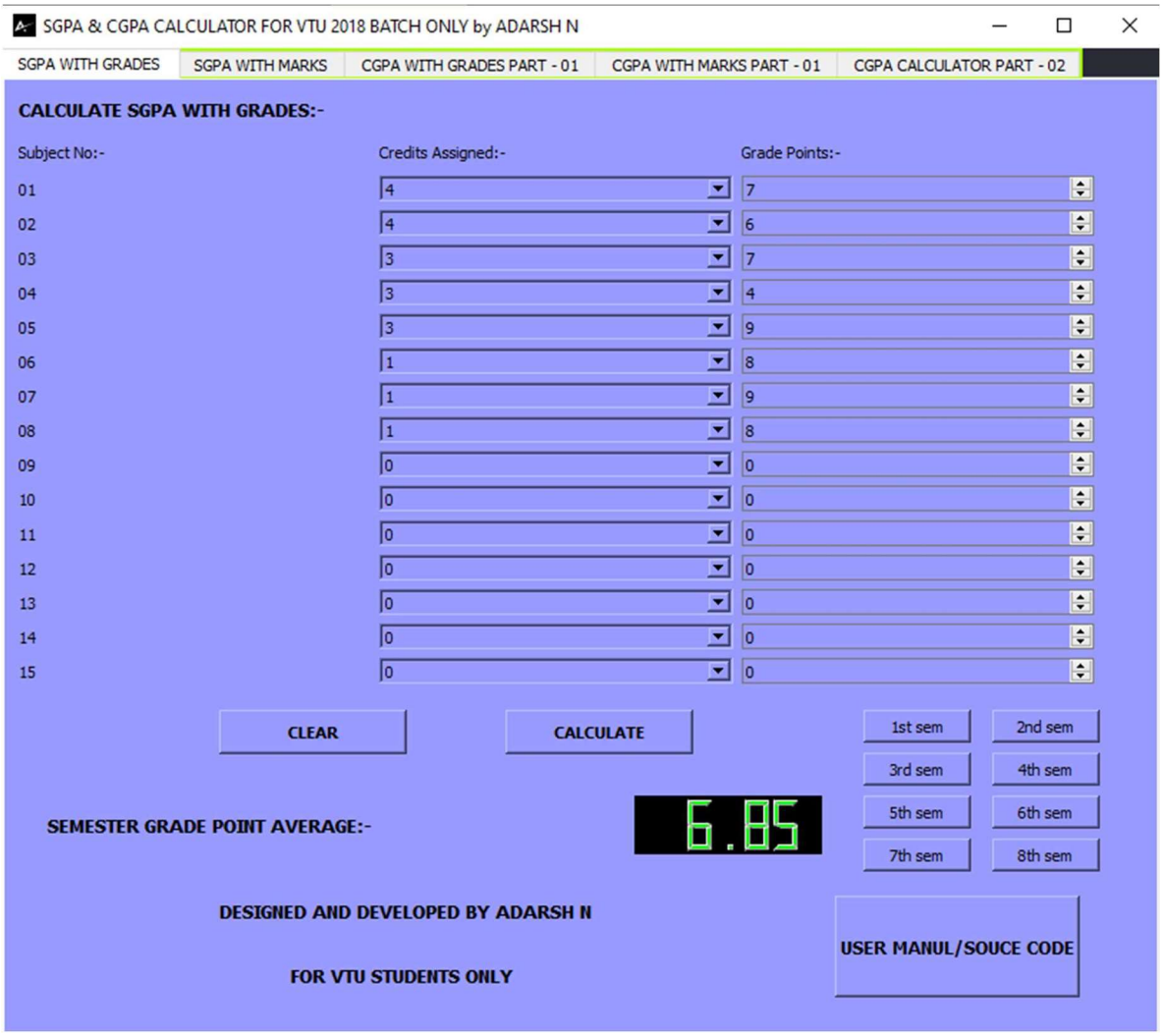
```
        msg.setIcon(QtWidgets.QMessageBox.Critical)
        msg.setText("Enter the values")
        msg.setInformativeText('More information')
        msg.setWindowTitle("Error")
        msg.exec_()

def calculate_cgpa_with_grades(self, lcdNumber_1):
    sum_of_ci_gi = 0
    sum_of_credits_earned = sum(credits_earned)
    try:
        for i in range(len(grade_points)):
            sum_of_ci_gi = sum_of_ci_gi + (credits[i] * grade_points[i])

        if lcdNumber_1 is True:
            self.lcdNumber_7.display(sum_of_ci_gi)
            self.lcdNumber_6.display(sum_of_credits_earned)
        elif lcdNumber_1 is False:
            self.lcdNumber_3.display(sum_of_ci_gi)
            self.lcdNumber_4.display(sum_of_credits_earned)
        credits.clear()
        grade_points.clear()

    except ZeroDivisionError:
            msg = QtWidgets.QMessageBox()
            msg.setIcon(QtWidgets.QMessageBox.Critical)
            msg.setText("Enter the values")
            msg.setInformativeText('More information')
            msg.setWindowTitle("Error")
            msg.exec_()

def check_grade(self, grade):
    if(grade!=0):
        if(grade<11):
            print(grade)
        else:
            msg = QtWidgets.QMessageBox()
            msg.setIcon(QtWidgets.QMessageBox.Critical)
            msg.setText("Enter a valid Grades ")
            msg.setInformativeText('More information')
            msg.setWindowTitle("Error")
            msg.exec_()

#convert marks to grades
def marks_to_grades():
        for marks in total_marks:
            if(marks>=90):
                grade_points.append(10)
            elif(marks<90 and marks>=80):
                 grade_points.append(9)
            elif(marks<80 and marks>=70):
                grade_points.append(8)
```

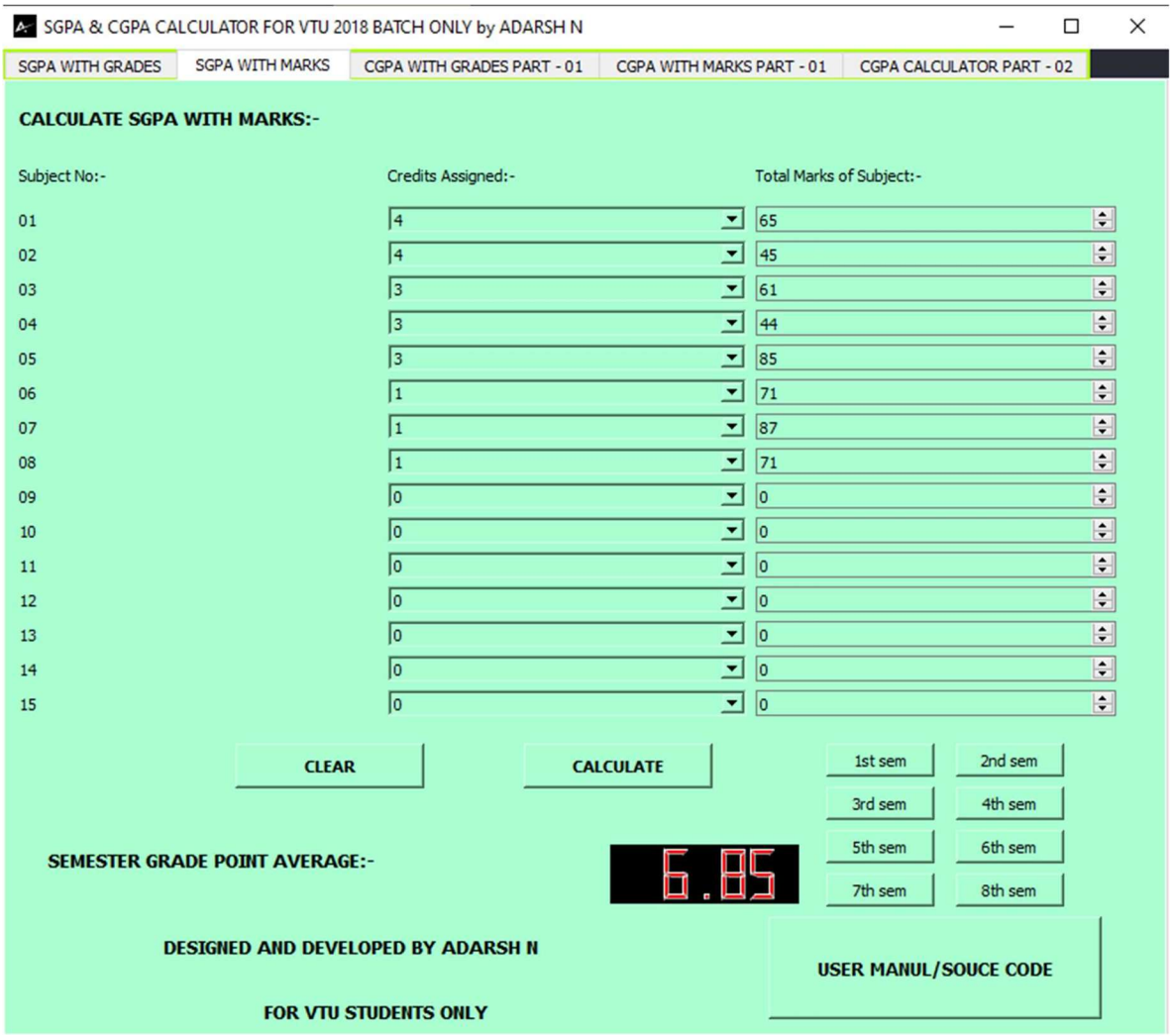
```
            elif(marks<70 and marks>=60):
                grade_points.append(7)
            elif(marks<60 and marks>=45):
                grade_points.append(6)
            elif(marks<45 and marks>=40):
                grade_points.append(4)
            elif(marks<40):
                grade_points.append(0)

    def calculate_with_grades(self, lcdNumber):
        sum_sgpa = 0
        sum_of_credits = sum(credits)
        try:
            for i in range(len(grade_points)):
                sum_sgpa = sum_sgpa + (credits[i] * grade_points[i])
            final_sgpa = sum_sgpa/sum_of_credits
            if lcdNumber is True:
                self.lcdNumber.display(final_sgpa)
            elif lcdNumber is False:
                self.lcdNumber_2.display(final_sgpa)
            credits.clear()
            grade_points.clear()
        except ZeroDivisionError:
                msg = QtWidgets.QMessageBox()
                msg.setIcon(QtWidgets.QMessageBox.Critical)
                msg.setText("Enter the values")
                msg.setInformativeText('More information')
                msg.setWindowTitle("Error")
                msg.exec_()

    if __name___== "__main__":
        import sys
        app = QtWidgets.QApplication(sys.argv)
        MainWindow = QtWidgets.QMainWindow()
        ui = Ui_MainWindow()
        ui.setupUi(MainWindow)
        MainWindow.show()
        sys.exit(app.exec_())
```

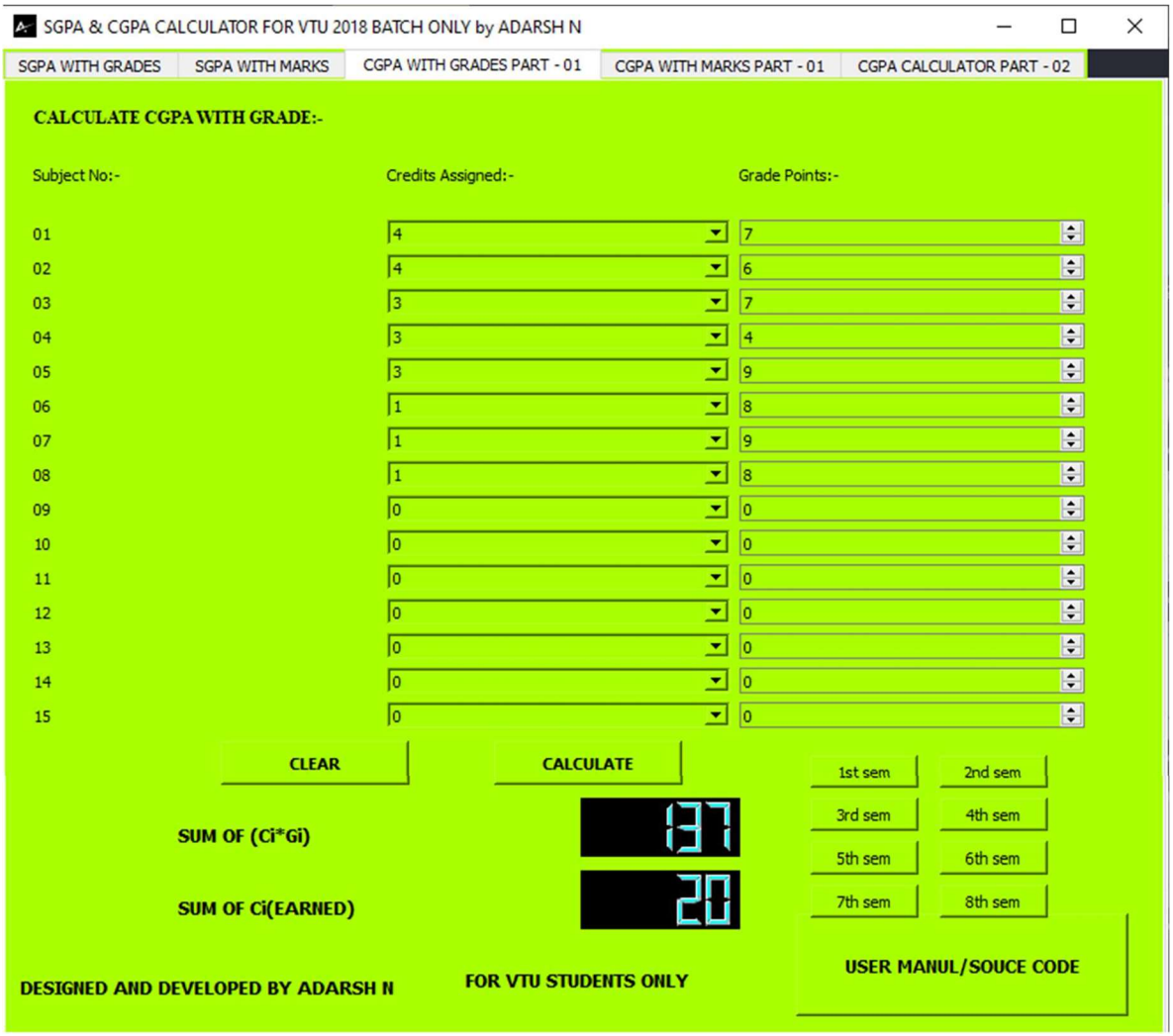
CHAPTER 8
SNAPSHOTS

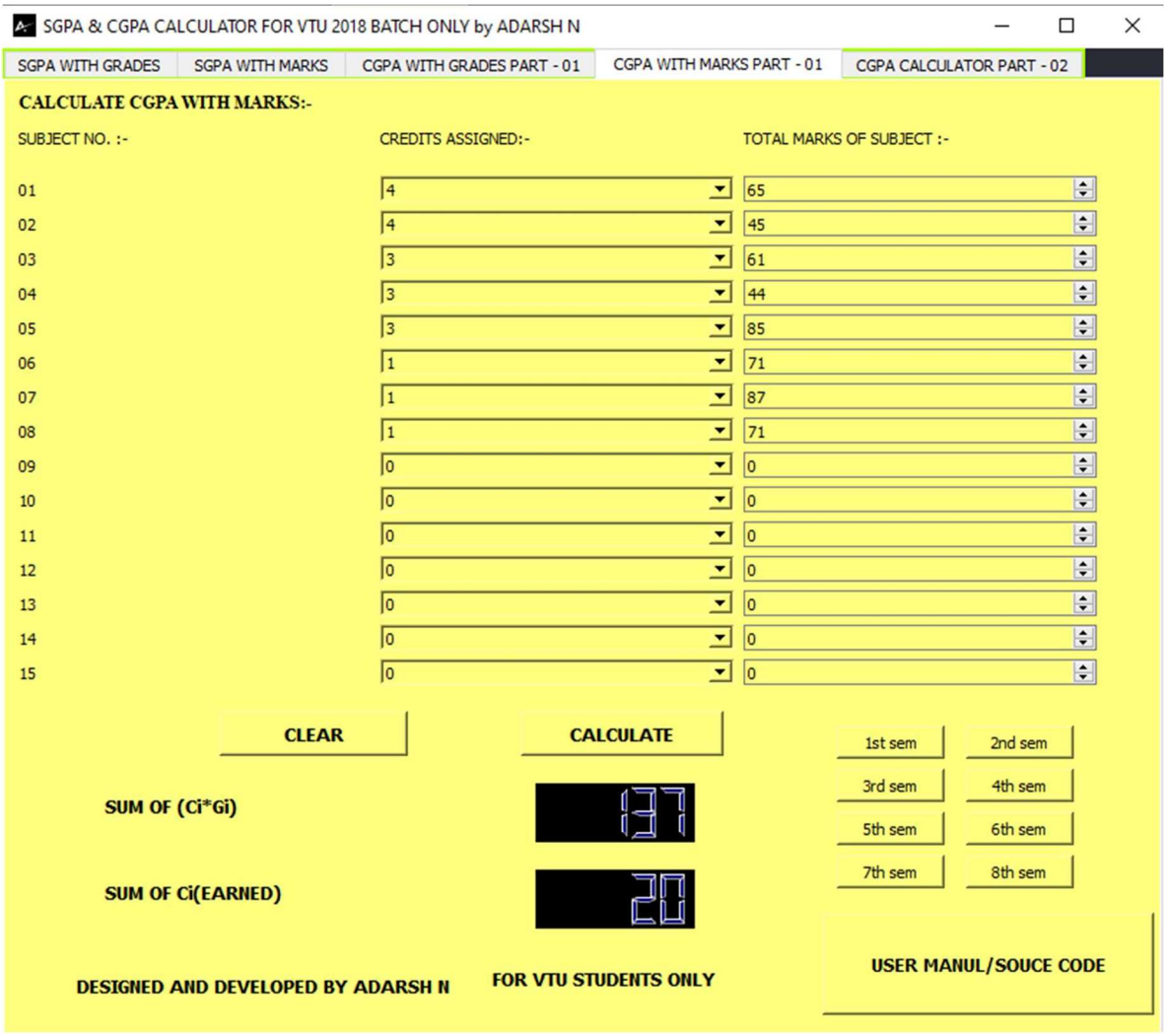
8. SNAPSHOT


SGPA & CGPA CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N
SGPA WITH GRADES
SGPA WITH MARKS
CGPA WITH GRADES PART - 01
CGPA WITH MARKS PART - 01
CGPA CALCULATOR PART - 02
CALCULATE SGPA WITH GRADES:-
Subject No:-
Credits Assigned:-
Grade Points:-
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
CLEAR
CALCULATE
1st sem
2nd sem
3rd sem
4th sem
5th sem
6th sem
7th sem
8th sem
SEMESTER GRADE POINT AVERAGE:-
DESIGNED AND DEVELOPED BY ADARSH N
FOR VTU STUDENTS ONLY
USER MANUL/SOUCE CODE


SGPA & CGPA CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N
SGPA WITH GRADES
SGPA WITH MARKS
CGPA WITH GRADES PART - 01
CGPA WITH MARKS PART - 01
CGPA CALCULATOR PART - 02
CALCULATE SGPA WITH GRADES:-
Subject No:-
Credits Assigned:-
Grade Points:-
01 4 7
02 4 6
03 3 7
04 3 4
05 3 9
06 1 8
07 1 9
08 1 8
09 0 0
10 0 0
11 0 0
12 0 0
13 0 0
14 0 0
15 0 0
CLEAR
CALCULATE
1st sem
2nd sem
3rd sem
4th sem
5th sem
6th sem
7th sem
8th sem
SEMESTER GRADE POINT AVERAGE:-
6.85
DESIGNED AND DEVELOPED BY ADARSH N
FOR VTU STUDENTS ONLY
USER MANUL/SOUCE CODE


SGPA & CGPA CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N
SGPA WITH GRADES
SGPA WITH MARKS
CGPA WITH GRADES PART - 01
CGPA WITH MARKS PART - 01
CGPA CALCULATOR PART - 02
CALCULATE SGPA WITH MARKS:-
Subject No:-
Credits Assigned:-
Total Marks of Subject:-
01 4 65
02 4 45
03 3 61
04 3 44
05 3 85
06 1 71
07 1 87
08 1 71
09 0 0
10 0 0
11 0 0
12 0 0
13 0 0
14 0 0
15 0 0
CLEAR
CALCULATE
1st sem
2nd sem
3rd sem
4th sem
5th sem
6th sem
7th sem
8th sem
SEMESTER GRADE POINT AVERAGE:-
6.85
DESIGNED AND DEVELOPED BY ADARSH N
FOR VTU STUDENTS ONLY
USER MANUL/SOUCE CODE


SGPA & CGPA CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N
SGPA WITH GRADES
SGPA WITH MARKS
CGPA WITH GRADES PART - 01
CGPA WITH MARKS PART - 01
CGPA CALCULATOR PART - 02
CALCULATE CGPA WITH GRADE:-
Subject No:-
Credits Assigned:-
Grade Points:-
01 4 7
02 4 6
03 3 7
04 3 4
05 3 9
06 1 8
07 1 9
08 1 8
09 0 0
10 0 0
11 0 0
12 0 0
13 0 0
14 0 0
15 0 0
CLEAR
CALCULATE
1st sem
2nd sem
3rd sem
4th sem
5th sem
6th sem
7th sem
8th sem
SUM OF (Ci*Gi)
137
SUM OF Ci(EARNED)
20
USER MANUL/SOUCE CODE
DESIGNED AND DEVELOPED BY ADARSH N
FOR VTU STUDENTS ONLY


SGPA & CGPA CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N
SGPA WITH GRADES
SGPA WITH MARKS
CGPA WITH GRADES PART - 01
CGPA WITH MARKS PART - 01
CGPA CALCULATOR PART - 02
CALCULATE CGPA WITH MARKS:-
SUBJECT NO. :-
CREDITS ASSIGNED:-
TOTAL MARKS OF SUBJECT :-
01 4 65
02 4 45
03 3 61
04 3 44
05 3 85
06 1 71
07 1 87
08 1 71
09 0 0
10 0 0
11 0 0
12 0 0
13 0 0
14 0 0
15 0 0
CLEAR
CALCULATE
1st sem
2nd sem
3rd sem
4th sem
5th sem
6th sem
7th sem
8th sem
SUM OF (Ci*Gi)
137
SUM OF Ci(EARNED)
20
USER MANUL/SOUCE CODE
DESIGNED AND DEVELOPED BY ADARSH N
FOR VTU STUDENTS ONLY


SGPA & CGPA CALCULATOR FOR VTU 2018 BATCH ONLY by ADARSH N
SGPA WITH GRADES
SGPA WITH MARKS
CGPA WITH GRADES PART - 01
CGPA WITH MARKS PART - 01
CGPA CALCULATOR PART - 02
ENTER THE DETAILS BELOW:-
1ST SEM SUM OF (Ci*Gi)
137
1ST SEM SUM OF Ci(EARNED)
20
2ND SEM SUM OF(Ci*Gi)
122
2ND SEM SUM OF Ci(EARNED)
20
3RD SEM SUM OF (Ci*Gi)
152
3RD SEM SUM OF Ci(EARNED)
24
4TH SEM SUM OF (Ci*Gi)
191
4TH SEM SUM OF Ci(EARNED)
24
5TH SEM SUM OF(Ci*Gi)
199
5TH SEM SUM OF Ci(EARNED)
25
6TH SEM SUM OF (Ci*Gi)
0
6TH SEM SUM OF Ci(EARNED)
0
7TH SEM SUM OF (Ci*Gi)
0
7TH SEM SUM OF Ci(EARNED)
0
8TH SEM SUM OF (Ci*Gi)
0
8TH SEM SUM OF Ci(EARNED)
0
CLEAR
CALCULATE
CGPA (CUMULATIVE GRADE POINT AVERAGE):-
7.088
DESIGNED AND DEVELOPED BY ADARSH N
FOR VTU STUDENTS ONLY
USER MANUL/SOUCE CODE

CHAPTER 9

CONCLUTION

9. CONCLUTION

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project:

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

10. REFERENCE

[1] Student Information Management Decision System Based on Decision Tree Classification Algorithm Wang Yanxia2022 IEEE 5th

International Conference on Information Systems and Computer Aided Education (ICISCAE) Year: 2022 | Conference Paper |

Publisher: IEEE

[2] Construction of intelligenceStudent Management Information System Platform Based on Big Data Analysis Xu Xing; Qigong

Zhang2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC) Year: 2022 | Conference Paper |

Publisher: IEEE

[3] Student Management Information System Based on Data mining Sinew Li2022 International Conference on Frontiers of Artificial

Intelligence and Machine Learning (FAIML) Year: 2022 | Conference Paper | Publisher: IEEE Analysis and Design of College

Student

[4] Award Management System WuNan Ding 2021 2nd International Conference on Computer Science and Management Technology

(ICCSMT)Year: 2021|-Conference Paper - Publisher: IEEE

[5] Research and Exploration of College Student Award Management System Based on Information System Under the Background of Big

DataWuNan Ding 2021 2nd International Conference on Big Data Economy and Information Management (BDEIM)Year: 2021 -

Conference Paper -Publisher: IEEE

[6] Intelligent Analysis of Students' Information Management in Higher Vocational Colleges in the Era of Big Data Management

(ICMEIM) Year: 2020 | Conference Paper | Publisher: IEEE

[7] Basic Project Management Documentation Based on the Example of the Student Project AGH Lean Line Katarzyna Sty; Jakub Licks;

Klaudia Drobek2019 8th International Conference on Industrial Technology and Management (ICITM) Year: 2019 | Conference Paper

| Publisher: IEEE

[8] Energy efficient all-digital phase locked loop architecture design on high resolution fast clocking time to digital converter (TDC) using

model prescient control (MPC) TM Sathish Kumar, PS Periasamy - Wireless Personal Communications, 2018 – Springer

[9] Research on College Students' Social Practice Management Based on Intelligent Workflow Gongli Luo; Faliang Nie; Junwei Du2011

International Conference of Information Technology, Computer Engineering and Management SciencesYear: 2011 | Volume: 4 |

Conference Paper | Publisher: IEEE

[10] S.R.Bharamagoudar et al , “Web-Based Student Information Management System ,International Journal of Advanced Research in

Computer and Communication Engineering Vol. 2, Issue 6, June 2013.

[11] Zhi-gang YUE,You-we JIN,“Thedevelopment and design of the student management system

based on

the network environment",2010 International Conference on Multimedia Communications,978-0-7695-4136-5/10 2010 IEEE. Joe M,

Shiau JK, Zhang HW. Application of Web Technologies in Automation Technology Education. International Journal of Computers

and Applications. 2009; 31:4.

[12] Xiang Fu, Boris Peltsverger, Kai Qian, Lixin Tao, Jigang Liu. APOGEE – Automated Project Grading and Instant Feedback System

for Web Based Computing, Computer Science and Information Technology, 2nd IEEE International Conference, 2009

[13] Ritvars Bregzis, Calvin Gotlieb, Carole Moore. The Beginning of Automation in the University of Toronto Library, 1963-1972, in

IEEE Annals of the History of Computing, 2002

[14] Prof. Godswill Obioma, Prof. Ismail Junaidu, Dr. Grace Ajagun. The Automation of Educational Assessment in Nigeria: Challenges

and Implications for Pre-service Teacher Education, 39th Annual Conference of the International Association for Educational

Assessment.

[15] Gerald Weber. Defining the Paperless Workplace with the Paper Metaphor-Not a Contradiction in Terms, Conference: Proceedings of

the Fourth Australasian Workshop on Health Informatics and Knowledge Management, 120.