# BMI CALCULATOR USING JAVA AND MYSQL

A Project Report
Submitted in partial fulfillment of the requirements for the
Award of Degree of (Bachelor of Computer Applications)

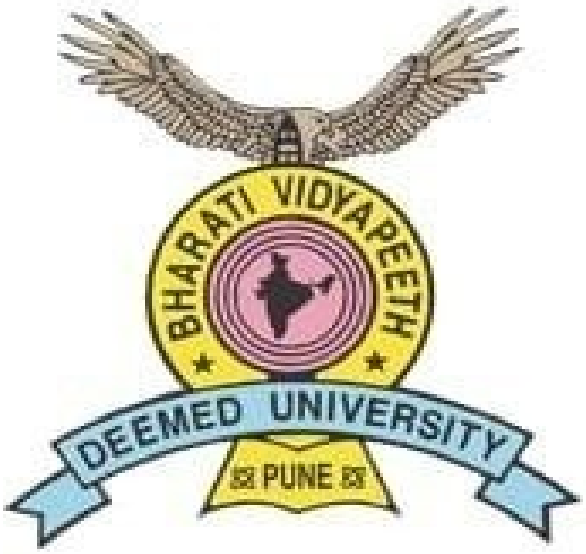
**2022-2024**


**Submitted by:**

**Prince Kumar**

**Guided by:**

**Shekhar Kundra**



**BHARATI VIDYAPEETH DEEMED UNIVERSITY SCHOOL OF DISTANCE EDUCATION**
**Academic Study Center - BVIMR, New Delhi An**
**ISO 9001:2008 Certified Institute**
**NAAC Accredited Grade "A" University**

## Student Undertaking

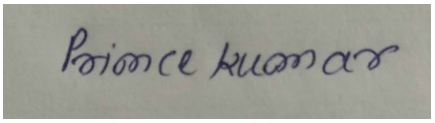
## Certificate of Originality

I Prince Kumar , Bachelors of Computer Application V$^{th}$ Sem would like to declare that the project report entitled BMI Calculator Submitted to Bharati Vidyapeeth University Pune, School of Distance Education Pune, Academic Study Centre BVIMR New Delhi in partial fulfillment of the requirement for the award of the degree.

It is an original work carried out by me under the guidance of Mr. Shekhar Kundra .

All respected guides, faculty member and other sources have been properly acknowledged and the report contains no plagiarism.
All respected guides, faculty member and other sources have been properly acknowledged and the report contains no plagiarism.

**PRINCE KUMAR**



**Student Name with Signature**

## Preface

In the preparation of this project of Accountancy, I have precisely demarcated all the important points. I have made my best possible efforts to remove all the errors.

It is a great pleasure for me to thank all those valuable suggestions thathave been given to me by Mr. Shekhar Kundra  Sir.

I must thank the almighty for this inspiration and guidance, teachers who directed me to complete this project file.

# Acknowledgement

I would like to express my special thanks of gratitude to my teachers Mr. Shekhar Kundra , for guidance regarding this project who gave me the golden opportunity to do this wonderful project of Computer Application I came to know about so many new things I am really thank full to them.

**PRINCE KUMAR**

**(Bachelors of Computer Application)**

# INDEX

**Chapter 3. SYSTEM DESIGN**



**Chapter 4. TESTING AND IMPLEMENTATION**



**Chapter 5 CONCLUSION AND REFERENCES**

# Chapter 1: INTRODUCTION

## 1.1 Introduction  about project

Body: -In contemporary society, health consciousness has become an integral part of daily life. With the rise in lifestyle-related diseases and a growing awareness of the importance of maintaining a healthy weight, individuals are increasingly seeking tools to monitor and manage their health effectively. Among these tools, the Body Mass Index (BMI) stands out as a simple yet powerful indicator of a person's weight status in relation to their height. The BMI has become a widely accepted measure used by healthcare professionals, researchers, and individuals alike to assess and track body composition and overall health. Recognizing the significance of BMI in health management, this project aims to design and develop an advanced BMI calculator with enhanced functionalities and user experience.

### 1.1.1 Background

The concept of BMI dates back to the early 19th century when Adolphe Queenlet, a Belgian mathematician, proposed the Queenlet Index as a means to assess the degree of obesity in populations. Over time, this index evolved into the Body Mass Index, a simple formula calculated by dividing an individual's weight (in kilograms) by the square of their height (in meters). The resulting numerical value categorizes individuals into different weight status categories, such as underweight, normal weight, overweight, and obese, providing a quick snapshot of their health status.

In recent decades, BMI has gained widespread popularity due to its

simplicity and effectiveness in assessing weight status. It serves as a valuable screening tool for identifying individuals at risk of various health conditions associated with excess weight, including cardiovascular diseases, diabetes, and certain cancers. Moreover, BMI is extensively used in clinical practice to monitor patients' progress during weight management interventions and to evaluate population-level trends in obesity rates.

However, despite its widespread use, BMI has its limitations. Critics argue that it does not account for variations in body composition, such as muscle mass and bone density, and may misclassify individuals, particularly athletes and older adults. Additionally, BMI does not provide insight into the distribution of body fat, which is an important determinant of health risk. Recognizing these limitations, researchers have developed alternative indices and tools to complement BMI assessment, such as waist circumference measurements and body fat percentage estimation.

### 1.1.2 Objectives

The primary objective of this project is to design and develop an advanced BMI calculator that addresses the limitations of traditional BMI assessment tools while providing additional features to enhance user experience and promote health awareness. The specific objectives include:

1. To incorporate advanced algorithms that account for variations in body composition, such as muscle mass and bone density, into the BMI calculation formula.
2. To integrate additional anthropometric measurements, such as waist circumference and hip circumference, to provide a more comprehensive assessment of body composition and distribution of body fat.
3. To design a user-friendly interface that allows individuals to input

their anthropometric data easily and interpret the results accurately.
4. To provide personalized recommendations and resources based on BMI results to promote healthy lifestyle choices and facilitate weight management.
5. To ensure the accuracy and reliability of the BMI calculator through rigorous testing and validation procedures.

By achieving these objectives, the proposed BMI calculator aims to offer a more accurate and informative tool for individuals to assess their weight status and make informed decisions about their health.

### 1.1.3 Methodology

The development of the advanced BMI calculator will involve a multi-stage process, encompassing algorithm design, software development, user interface design, and testing/validation. The methodology can be outlined as follows:

1. Algorithm Design: This stage will involve reviewing existing BMI calculation formulas and exploring advanced algorithms that account for variations in body composition. Collaborating with experts in the fields of nutrition, physiology, and biostatistics will help refine the algorithm to improve accuracy and reliability.

2. Software Development: Once the algorithm is finalized, the software development team will translate the algorithm into computer code using appropriate programming languages and tools. The BMI calculator will be designed as a web-based application, accessible through desktop and mobile devices, to maximize usability and accessibility.

3. User Interface Design: A user-centered approach will be adopted to design the interface of the BMI calculator, focusing on simplicity, clarity, and interactivity. Wireframing and prototyping techniques will be

utilized to visualize the layout and functionality of the application, with feedback obtained from target users to guide refinements.

4. Integration of Additional Features: In addition to BMI calculation, the calculator will incorporate features such as waist circumference measurement, hip circumference measurement, and body fat percentage estimation. These additional measurements will be seamlessly integrated into the interface to provide users with a comprehensive assessment of their body composition.

5. Testing and Validation: The final stage of the development process will involve rigorous testing and validation of the BMI calculator to ensure accuracy, reliability, and usability. This will include conducting simulated testing with synthetic data, as well as real-world testing with volunteer participants representing diverse demographics.

### 1.1.4 Expected Outcomes

Upon completion of the project, several key outcomes are anticipated:

1. An advanced BMI calculator with enhanced accuracy and additional features, including waist circumference measurement and body fat percentage estimation.
2. A user-friendly interface that simplifies the process of inputting anthropometric data and interpreting the results.
3. Personalized recommendations and resources based on BMI results to promote healthy lifestyle choices and facilitate weight management.
4. Validation of the BMI calculator through rigorous testing procedures, ensuring its accuracy and reliability across diverse populations.
5. Dissemination of the BMI calculator through various channels, including web-based platforms, mobile applications, and healthcare settings, to reach a wide audience and promote health awareness.

## 1.2 Need of computerization of system

1. Automation and Accuracy: Computerization allows for the automation of BMI calculations, reducing the risk of human error inherent in manual calculations. Algorithms can be programmed to perform calculations consistently and accurately, ensuring precise BMI values.

2. Efficiency and Speed: By computerizing the BMI calculation process, results can be obtained swiftly, enabling users to assess their weight status promptly. This is particularly important in healthcare settings where quick access to BMI information is crucial for patient evaluation and treatment planning.

3. Integration of Additional Measurements: Computerized BMI calculators can easily incorporate additional anthropometric measurements, such as waist circumference and hip circumference, to provide a more comprehensive assessment of body composition. This integration enhances the utility of the BMI calculator, offering users a broader perspective on their health status.

4. Accessibility and User-Friendliness: Computerized BMI calculators can be designed with user-friendly interfaces accessible through various platforms, including desktop computers, laptops, tablets, and smartphones. This accessibility ensures that individuals can conveniently access the BMI calculator whenever needed, promoting widespread use and adoption.

5. Data Storage and Analysis: Computerized systems facilitate the storage and analysis of BMI data, allowing for the tracking of individuals' weight status over time. This longitudinal data can be valuable for monitoring trends in population health and evaluating the effectiveness of interventions aimed at promoting healthy weight management.

6. Customization and Personalization: Computerized BMI calculators can be tailored to provide personalized recommendations and resources based on users' BMI results. This customization enhances user engagement and motivation, empowering individuals to make informed decisions about their health and well-being.

## 1.3 Proposed Software

1. Platform and Compatibility: The BMI calculator software will be developed as a web-based application to ensure compatibility across various devices and operating systems, including desktop computers, laptops, tablets, and smartphones. This platform-agnostic approach will maximize accessibility, allowing users to access the BMI calculator conveniently from any device with internet connectivity.

2. User Interface Design: The user interface of the BMI calculator software will be intuitively designed with a focus on simplicity, clarity, and interactivity. Users will be guided through the BMI calculation process with clear instructions and input fields for entering their height and weight measurements. Visual aids, such as sliders or drop-down menus, may be incorporated to facilitate data entry and improve user engagement.

3. Advanced Algorithms: The BMI calculator software will utilize advanced algorithms to calculate BMI accurately, considering variations in body composition, such as muscle mass and bone density. These algorithms will be based on established mathematical formulas and may be refined through collaboration with experts in the fields of nutrition, physiology, and biostatistics to ensure accuracy and reliability.

4. Additional Measurements: In addition to BMI calculation, the software will integrate additional anthropometric measurements, such as waist circumference and hip circumference, to provide a more comprehensive assessment of body

composition and distribution of body fat. Users will have the option to input these measurements alongside their height and weight to receive a more detailed analysis of their health status.

5. Personalized Recommendations: Based on the calculated BMI and additional measurements, the software will generate personalized recommendations and resources to promote healthy lifestyle choices and facilitate weight management. These recommendations may include dietary guidelines, exercise recommendations, and links to relevant educational materials or support resources tailored to the user's specific health profile.

6. Privacy and Security: The BMI calculator software will prioritize user privacy and data security by implementing robust encryption protocols and compliance with data protection regulations. Users' personal health information will be handled confidentially, and measures will be in place to safeguard against unauthorized access or data breaches.

7. Testing and Validation: Prior to deployment, the BMI calculator software will undergo rigorous testing and validation procedures to ensure accuracy, reliability, and usability. This will include simulated testing with synthetic data, as well as real-world testing with volunteer participants representing diverse demographics. Feedback from testing will be used to refine and optimize the software before its release to the public.

## 1.4 Important of The Work

1. Health Monitoring and Assessment: BMI serves as a valuable metric for assessing an individual's weight status in relation to their height. By providing a numerical value that categorizes individuals into different weight status categories (underweight, normal weight, overweight, obese), a BMI calculator allows for quick and easy monitoring of health status. Regular BMI assessments can help individuals and healthcare professionals track changes in weight over time and identify potential health risks associated with underweight, overweight,

or obesity.

2. Early Detection of Health Risks: BMI calculation can aid in the early detection of health risks associated with weight-related conditions such as cardiovascular diseases, diabetes, hypertension, and certain cancers. Individuals with elevated BMI levels may be at higher risk for these conditions, making early detection and intervention crucial for preventing or managing chronic diseases.

3. Health Promotion and Education: BMI calculators play a pivotal role in promoting health awareness and education. By providing individuals with information about their weight status and associated health risks, BMI calculators empower them to make informed decisions about their lifestyle choices, including diet, exercise, and overall health management. Additionally, BMI calculators can serve as educational tools in healthcare settings, schools, workplaces, and community outreach programs to raise awareness about the importance of maintaining a healthy weight.

4. Clinical Decision Making: In clinical practice, BMI calculation is often used by healthcare professionals to assess patients' overall health status and guide clinical decision-making. BMI serves as a screening tool for identifying individuals who may benefit from further assessment and intervention, such as dietary counseling, physical activity recommendations, or weight management programs. BMI calculators provide healthcare providers with a standardized method for evaluating patients' weight status and monitoring changes over time.

5. Population Health Surveillance: BMI calculators contribute to population health surveillance efforts by providing data on trends in obesity and overweight prevalence at the population level. This information is essential for policymakers, public health officials, and researchers to monitor changes in population health over time, identify high-risk groups, and develop targeted interventions to address obesity-related health disparities.

6. Research and Epidemiology: BMI calculators are valuable tools in research and epidemiological studies investigating the associations between weight status and various health outcomes. Researchers use BMI data to examine trends in obesity prevalence, explore risk factors for weight-related diseases, and evaluate the effectiveness of interventions aimed at promoting healthy weight management. BMI calculators facilitate standardized data collection and analysis, enabling researchers to draw meaningful conclusions from large-scale studies.

# Chapter 2: SYSTEM ANALYSIS

## 1.1 Feasibility Study of s/w includes its types

1. Technical Feasibility:

Algorithm Development: Assess the technical feasibility of developing accurate algorithms for BMI calculation, including considerations for accounting for variations in body composition.

Software Development Tools: Evaluate the availability and suitability of programming languages, frameworks, and development tools for implementing the BMI calculator software.

integration of Additional Measurements: Determine the technical feasibility of integrating additional anthropometric measurements, such as waist circumference and hip circumference, into the software.

2. Economic Feasibility:

Cost-Benefit Analysis: Conduct a cost-benefit analysis to determine whether the benefits of developing the BMI calculator software outweigh the costs involved in terms of development, maintenance, and potential revenue generation.

Resource Availability: Assess the availability of financial resources, human resources (e.g., developers, designers), and technological infrastructure required for software development.

Market Demand: Analyze the market demand for BMI calculator software and potential revenue streams, such as licensing fees, subscription models, or advertising revenue.

3. Operational Feasibility:

User Requirements: Gather user requirements through surveys, interviews, or focus groups to understand user needs and preferences for the BMI calculator software.

User Interface Design: Assess the feasibility of designing a user-friendly interface that is intuitive, accessible, and engaging for a diverse user base.

Training and Support: Evaluate the feasibility of providing training and support to users, including documentation, tutorials, and customer service channels.

4. Scheduling Feasibility:

Timeline and Milestones: Develop a project timeline with clearly defined milestones for each phase of software development, including design, implementation, testing, and deployment.

Resource Allocation: Allocate resources effectively to ensure that the project stays on schedule and within budget constraints.

Risk Management: Identify potential risks and challenges that may impact the project timeline and develop contingency plans to mitigate these risks.

Based on the feasibility study, the types of BMI calculator software that may be considered include:

1. Basic BMI Calculator: This type of calculator provides a simple interface for users to input their height and weight measurements and calculates their BMI using a standard formula. It may offer basic features such as BMI classification and interpretation.

2. Advanced BMI Calculator: An advanced BMI calculator incorporates additional measurements, such as waist circumference and hip circumference, to provide a more comprehensive assessment of body composition and health status. It may also include personalized recommendations based on BMI results.

3. Mobile App: A mobile app version of the BMI calculator allows users to access the tool conveniently from their smartphones or tablets. It may offer features such as data synchronization across devices, reminder notifications, and integration with health tracking apps.

4. Web-Based Platform: A web-based BMI calculator accessible through a browser offers flexibility and accessibility for users on various devices. It may include interactive features, visualizations, and educational resources to enhance user engagement.

5. Clinical BMI Calculator: This type of calculator is designed for use in clinical settings by healthcare professionals for patient assessment and monitoring. It may include additional features such as patient management tools, **e**lectronic health record integration, and customizable reporting capabilities.

### 1.1.1 Types of Feasibility Study:

1. Technical Feasibility: This study assesses whether the technology required for the IBM calculator is available or feasible to develop. It examines aspects like hardware requirements, software capabilities, and any technological challenges that might arise.

2. Financial Feasibility: Financial feasibility looks at whether the development and production costs of the IBM calculator can be justified by its potential benefits and projected revenue. It involves analyzing the initial investment, ongoing expenses, potential profits, and return on investment (ROI).

3. Market Feasibility: This study evaluates the demand for the IBM calculator in the market. It involves researching potential customers, competitors, market trends, and any regulatory or legal factors that might

affect its success.

4. Operational Feasibility: Operational feasibility examines whether the IBM calculator can be effectively integrated into existing systems or processes. It considers factors like ease of use, compatibility with other devices or software, and the impact on workflow.

5. Schedule Feasibility: This study assesses whether the IBM calculator can be developed and brought to market within a reasonable timeframe. It involves creating a project timeline, identifying potential bottlenecks or delays, and mitigating risks to ensure timely delivery.

6. Environmental Feasibility: Environmental feasibility evaluates the potential environmental impact of the IBM calculator throughout its lifecycle. This includes considerations such as energy consumption, materials sourcing, manufacturing processes, and end-of-life disposal or recycling.

### 1.1.2 Software Development Life Cycle:

1. A system development life cycle is a logical process by which system analysts, software engineers, programmers, and end users build informationsystems and computer applications to solve business problems and needs.
2. The major phases involved in the MIS development process are referred toas system development life cycle. Each phase of the development process must have well defined objectives,

and at the end of each phase, progress towards meeting the objectives must be evaluated.

3. The development process should not continue until the objectives of all priorphases have been met.

4. System development life cycle is a phased approach to analysis and designto ensure that systems are best developed.

## 1.2 Analysis Methodology (Types)

1. Statistical Analysis: This methodology involves analyzing a large dataset of body measurements (height and weight) along with corresponding BMI calculations to derive statistical parameters such as mean, median, mode, standard deviation, and percentile ranges. This helps ensure that the BMI calculator produces results that align with typical population distributions.

2. Regression Analysis: Regression analysis can be used to establish a mathematical relationship between height, weight, and BMI. By fitting a regression model to the data, the calculator can predict BMI values based on input height and weight, allowing for interpolation between known data points.

3. Clinical Validation: This methodology involves comparing BMI calculations from the calculator against measurements taken by healthcare professionals using standardized methods. Clinical validation ensures that the calculator produces results consistent with established medical standards and guidelines.

4. User Testing: User testing involves gathering feedback from individuals who use the BMI calculator. This could include surveys, interviews, or usability testing to evaluate factors such as ease of use, clarity of instructions, and user satisfaction. User testing helps identify any issues with the calculator interface or functionality.

5. Cross-Validation: Cross-validation is a technique used to assess the performance of the BMI calculator by splitting the dataset into multiple subsets. The model is trained on one subset and tested on the remaining subsets, allowing for an evaluation of its predictive accuracy and generalization to new data.

6. Sensitivity Analysis: Sensitivity analysis involves examining how changes in input parameters (such as height and weight) affect the output (BMI calculation). This helps identify which inputs have the greatest impact on the results and can be used to assess the robustness of the calculator under different scenarios.

7. Comparative Analysis: Comparative analysis involves comparing the BMI calculator against other established calculators or methods for calculating BMI. This could include assessing differences in calculation algorithms, user interfaces, and accuracy of results.

### 1.2.1 System Analysis

1. User Requirements Gathering: This phase involves identifying the needs and preferences of the target users for the BMI calculator. It includes gathering information on user demographics, usability preferences, and any specific features or functionalities they require.

2. Functional Requirements Specification: Functional requirements define what the BMI calculator should do. This includes specifying input parameters (such as height and weight), calculation algorithms (such as the formula for BMI calculation), and output formats (such as displaying the calculated BMI value along with interpretation).

3. Non-Functional Requirements Specification: Non-functional requirements specify qualities that the BMI calculator system should have, such as performance, reliability, usability, security, and scalability. For example, the calculator should provide accurate results within a reasonable time frame and ensure user data privacy.

4. System Architecture Design: System architecture design involves defining the structure of the BMI calculator system, including its components, modules, and their interactions. It may include decisions about the programming language, database management system, user interface design, and integration with other systems or platforms.

5. Data Modeling: Data modeling involves designing the structure of the data used by the BMI calculator, including the representation of user inputs, intermediate calculations, and output results. This may involve creating entity-relationship diagrams, data flow diagrams, or schema designs for databases.

6. User Interface Design: User interface (UI) design focuses on creating an intuitive and user-friendly interface for the BMI calculator. It involves designing input forms for height and weight entry, displaying the calculated BMI value, providing clear instructions and error messages, and

ensuring accessibility across different devices and screen sizes.

7. Testing and Validation: Testing and validation involve verifying that the BMI calculator functions correctly and meets its requirements. This includes unit testing individual components, integration testing the system as a whole, and user acceptance testing to ensure it meets user expectations.

8. Documentation and Training: Documentation includes creating user manuals, technical documentation, and training materials to help users understand how to use the BMI calculator effectively. This ensures that users can make accurate measurements and interpret the results correctly.

## Algorithm:

Step: 1. Start

Step: 2. Define the BMI Calculator class with the following attributes:

- ✓ height (double)
- ✓ weight (double)
- ✓ Name (String)
- ✓ Prn (String)
- ✓ Course (String)
- ✓ Gmailid (String)
- ✓ Phone (String)
- ✓ Age (int)

- ✓ users (Map<String, String>)
- ✓ loggedInUser (String)

Step: 3. Define the BMI Calculator constructor to initialize height and weight.

Step: 4. Define methods for calculating BMI, BMI prime, and displaying results.

Step: 5. Define a method to generate a random password.

Step: 6. Define a method to clear the console screen.

Step: 7. Define the main method:

1. Initialize Scanner object for user input.
2. Initialize BMI Calculator object.
3. Create a loop for the BMI calculator menu:

a. If no user is logged in:

i. Display registration, login, and exit options.

ii. Based on user input:

- Perform user registration.
- Perform user login.
- Exit the program.

b. If a user is logged in:

i. Display BMI calculation and logout options.

ii. Based on user input:

- Calculate BMI.

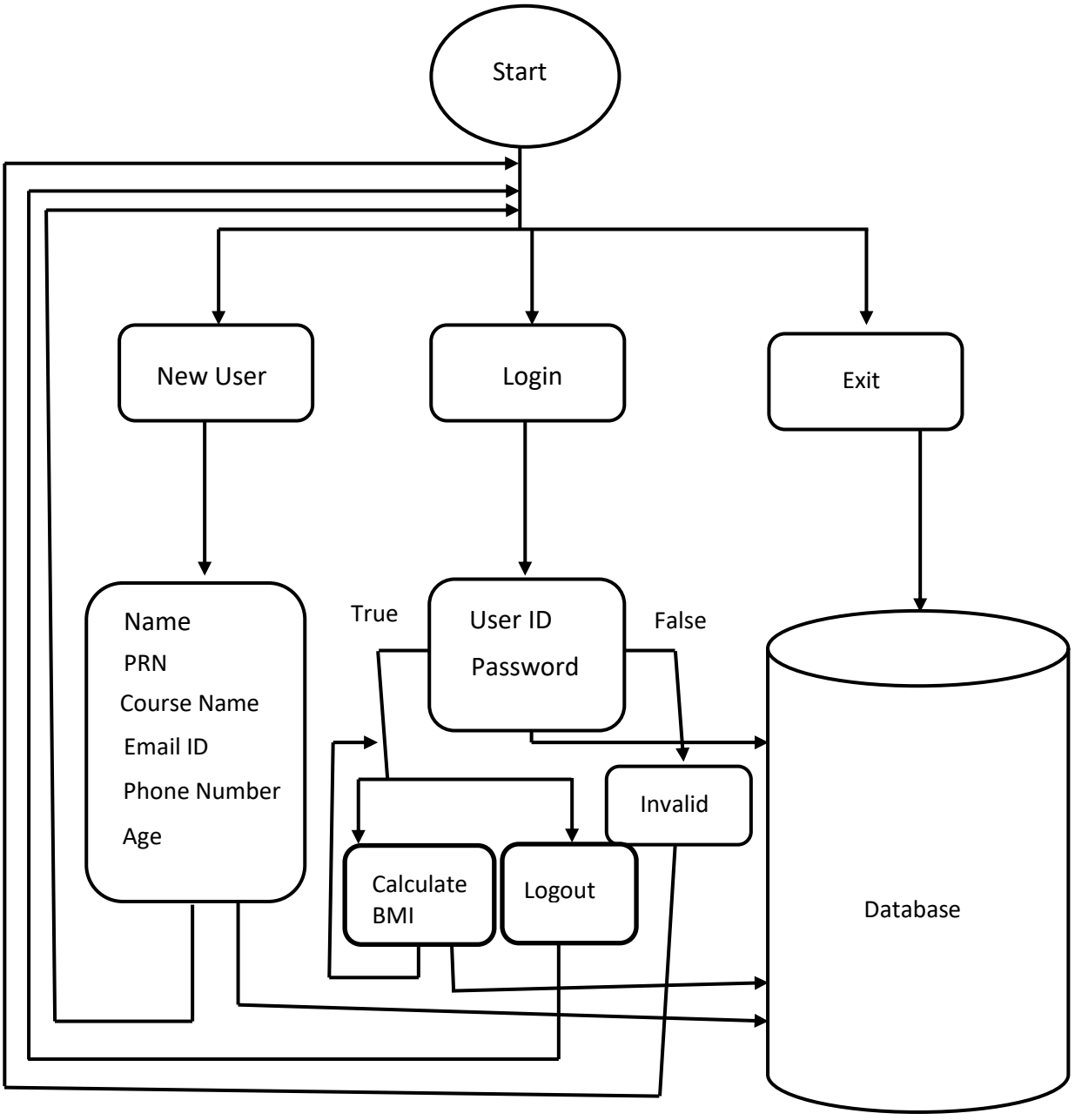
- Logout the user.

4. End loop.

Step: 8. End

## Flowchart

### Flowchart for Calculate BMI



## For New User Registration / Login / Exit

Step 1: Start

Step 2: Select 1 for New User Registration and 2 for Login and 3 for exit.

Step 3: If 1 is selected, then New Registration and go to Step 2.

Step 4: If 2 is selected, then Login it after verification and go to step 2.

i. Calculate BMI
ii. Logout

Step 5: If i. is selected, Calculate BMI and go to step 4.

Step 6: If ii. is selected, then Logout it after verification and go to step 2.

Step 7: If 3 is selected, then exist it and go to step 8.

Step 8: Stop

## 1.3 Choice of Platforms s/w & h/w

Many platforms have been designed to support mixed SW/HW implementations, but all of them suffer from the fact that there is no easy procedure capable to seamlessly plug hardware modules described in HDL to a pure software algorithm. Either the memory management is a burdensome task or the call of the hardware module is done by an embedded processor on the platform. Environments which support HW/SW implementations are generally based on a platform containing an embedded processor and some dedicated hardware logic like FPGA as described in the work of Andreas Koch [2]. The control program lies in the embedded processor. However, data on the host are available easily thanks to virtual serial ports. But the plugging of hardware modules inside the reference software running on the host remains the most difficult task. The work of Martyn Edwards and Benjamin Foard [3] is interesting in the way a FPGA-based algorithm can be activated from the host processor. This platform is based on the Calexico RC1000- PP board and communicates with the host by using the PCI bus. The control program is on the host processor, sends control information to the FPGA and transfers data in small shared memory which is part of the hardware platform. In this case, the designer/programmer must do explicitly the data transfer between the host and the local memory. Many other works about coprocessors have been reported in literature. Some examples are given in [4] [5]. However, the problem of seamless plug-in of HDL modules is still there, the data transfers in the charge ofthe designer with can be a very burdensome task when dealing with complex data-dominated video or multimedia algorithms.

## 1.4 Software used

A visual studio code is a lightweight software application with a powerful source code editor that runs on the desktop. It is a free source code editor developed by Microsoft for Windows, Mac OS and Linux. It is a software editor that has a rich extension of various languages like C, C++, Java, Python, PHP, Go, etc. and runtimelanguage extensions such as .NET and Unity. It is easy to edit, build, syntax highlighting, snippets, code refactoring and debugging. In visual studio code, we can change the application's background theme, keyboard shortcuts set on our preferences, install an extension and add additional functionality.

## 1.5 Hardware used

1. Application Servers:
These servers run the analytics applications and perform computations. They need sufficient processing power and memory to handle complex analytical tasks.

2. Storage Area Network (SAN):
For high-performance and scalable storage needs, a SAN may be used. SANs provide centralized and shared storage that can be accessed by multiple servers simultaneously. Another option for shared storage, NAS systems connect to the network and can be accessed by multiple servers.

3. Switches and Routers:
High-speed and reliable networking equipment is essential for data transfer between servers, storage systems, and other components in the analytics infrastructure.

4. CPU (Central Processing Unit):

Servers used in analytics systems typically require powerful CPUs to handle the computational load associated with data processing and analysis. For certain types of analytics, especially those involving machine learning and parallel processing, GPUs can significantly accelerate computations.

5. High-capacity RAM:

Analytics processes often involve manipulating and analyzing large datasets, requiring sufficient RAM to handle the data in memory.

6. Hard Disk Drives (HDD) or Solid-State Drives (SSD):

These are used for storing data. SSDs are often preferred for their faster read and write speeds, which can improve overall system performance.

7. Tape Libraries or Backup Servers:

To ensure data integrity and availability, backup systems are crucial. Tape libraries or backup servers are used to store copies of important data.

8. Monitoring Servers:

Tools for monitoring system performance and resource utilization.

9. Security Appliances:

Hardware devices such as intrusion detection/prevention systems and security gateways to protect the analytics infrastructure from cyber threats.

10. Uninterruptible Power Supply (UPS):

To ensure continuous power in case of outages. Servers generate heat, andproper cooling systems are essential to prevent overheating Client Devices.

# Chapter 3: SYSTEM DESGIN

## 1.1 Design Methodology

Design is the first step into the development phase for any engineered productor system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel.

System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. System Design for tech interviews is something that can’t be ignored! Almost every IT giant whether it be Facebook, Amazon, Google, Apple or any other ask various questions based on System Design concepts such as scalability, load-balancing, caching, etc. in the interview. This specifically designed System Design tutorial willhelp you to learn and master System Design concepts in the most efficient way

from basics to advanced level.

system design refers to the process of defining the architecture, modules, interfaces, data for a system to satisfy specified requirements. It is a multi-disciplinary field that involves trade-off analysis, balancing conflicting requirements, and making decisions about design choices that will impact the overall system.

## 1.2 Database Design

The database, called a Bmi_Records_Data, will have One table, one called BmiCalculateData. Each will hold information about BmiCalculateData. The One table will be linked through a foreign key. The customertable has the following fields:

| | |
|---|---|
| New User Registration | Name, PRN, Course Name, Email ID, Phone Number , Age |
| Login | UserId , Password |
| BMI Calculate | Enter Weight in Kg ,Enter Height in Meters |
| Logout | Step 2 |
| Exit | =>>>> |

## 1.3 Screen Design

Screen design refers to the graphic design and layout of user interfaces on displays. It is a sub-area of user interface design but is limited to monitors and displays.

In screen design, the focus is on maximizing usability and user experience by making user interaction as simple and efficient as possible. Technical background functions do not matter in screen design. Web design, on the other hand, is the process of creating websites that implement the functions not considered in screen design. You could say that screen design thus makes the functionality specified in the context of web design applicable to users.

In the days of terminals connected to remote computers, these screens were synonymous with physical screens—that is, presenting a new screen meant repainting the entire physical screen. As graphical user interface (GUI) systems have become common, screen design has taken on a somewhat different meaning. Today, in GUI applications, most people use the term “screen design “to refer to the design of a specific window or dialog box rather than the design of theentire physical screen. Consequently, the chapter focuses on the design of these individual frames of information. A variety of metrics and several

systems have been developed to help in developing effective screens. These systems have grown to encompass increasingly rich definitions of the visual characteristics of interfaces as well as the tasks they are used for. They promise eventually leading to a fully-featured screen designer's workbench” in which a designer could iteratively refine screen designs automatically generated from basic interface requirements and task descriptions

### 1.3.1 Applications of screen design:

Screen design is used in mobile and desktop versions of websites, apps, and any graphical user interface displayed on monitors or displays that enable users to interact with electronic devices. This includes computers, portable media players, gaming devices, smartphones, and household, office, or industrial controls.

What makes good screen design? Focus on users
Good screen design always focuses on users and takes their needs and expectations into account when designing a user interface. all elements (text, images, icons, and buttons) are arranged and designed in a way that enables consumers to easily understand and apply them.

User-oriented screen design is based on knowing users' preferences, skills, and goals and requires a designer who can empathize with their audience to create simple and effective solution that is easy to execute. Thereby he must strike a careful balance between what is graphically appealing and what is practical for users.

### 1.3.2 Screen design doesn't reinvent the wheel

Good screen design considers that over the year’s users have become accustomedto certain elements and their function and appearance. People will accept and usenew applications more quickly if their design is similar to applications they are already familiar with. Therefore, some companies are developing a library of standard screen templates that developers can use. This approach not only helps enforce design standards but also accelerates screen design development.

## 1.4 Report design

Report Designer provides a graphical interface in which you can define data sources, datasets and queries, report layout positions for data regions and fields, and interactive features such as parameters and sets of reports that work together.

Report Designer is the heart of reporting tool providing a guided flow for report creation and styling. A complete reporting tool has an easily accessible and intuitive report designer that requires minimal training to use. Report Designer provides a graphical interface where developers or users can define data sources, datasets and queries, report layout positions for different data fields and interactive features as parameters and sets of reports that work together, etc. Thedesigner typically provides numerous styling capabilities, such as conditional formatting, CSS-like styling and countless wizards to help users create complex report layouts, style reports, manipulate data and build visually appealing presentations. By using data-bound items like maps, charts, crosstabs and sub reports, users can present interactive report decks and other elements (dashboards, reports by region, invoices, inventory reports, barcode reports and much more. There are different types of report designers – desktop, web-based, etc.

Desktop report designer enables developers and users to start from scratch with easy drag and drop design or use the huge library with pre-defined templates. It gives the opportunity to create, design, edit, preview, save, export and print reports seamlessly and without any code.

Web report designer has easy-to-use, fine-graded drag-and-drop functionality to layout and re-parent all items and sections. It features most functionalities of desktop report designers but adding more. One of its main benefits laid in enabling developers and report authors to bring report editing functionality to their end-users right from their web applications.

Visual Studio report designer targets developers that prefer VS for report authoring. It works with .NET type report definitions (.CS or .VB files). VS reportdesigner enables adding and editing of charts, tables and groups as well filling them with data, styling, sharing and export for ready reports.

If you need a full-functional reporting tool, try Telerik Reporting.

Telerik Reporting is a powerful easy-to-use embedded .NET reporting solution for web and desktop applications that supports: ASP.NET Core, Blazer, ASP.NETMVC, ASP.NET AJAX, HTML5, Angular, React, Vue, WPF, WinForms. Reporting enables creation, styling, viewing and export of rich, interactive and reusable reports to attractively present analytical and any business data. The reports can be added toany business application through report viewer controls and exported in more than 15 formats.

You can evaluate and see all the benefits of Telerik Reporting by downloading 30-day free trial. Doing so will get you access to our support with response time of less than 24 hours. The trial will also provide you with detailed resources to make the Getting started experience as smooth as possible.

### 1.4.1 Planning the report:

Before creating a report, identify the information that you want the report to provide and decide how to present that information. It is important to think through these details, then draw a mock-up on paper, which you use to get feedback from your report users. Most people cannot visualize what a report could be without a paper and pencil sketch. Planning saves time in the long run because you do not waste time creating a polished report that contains the wronginformation or layout. More frequently, you discover in this review process that the customer wants much more and can now articulate those requirements more successfully.

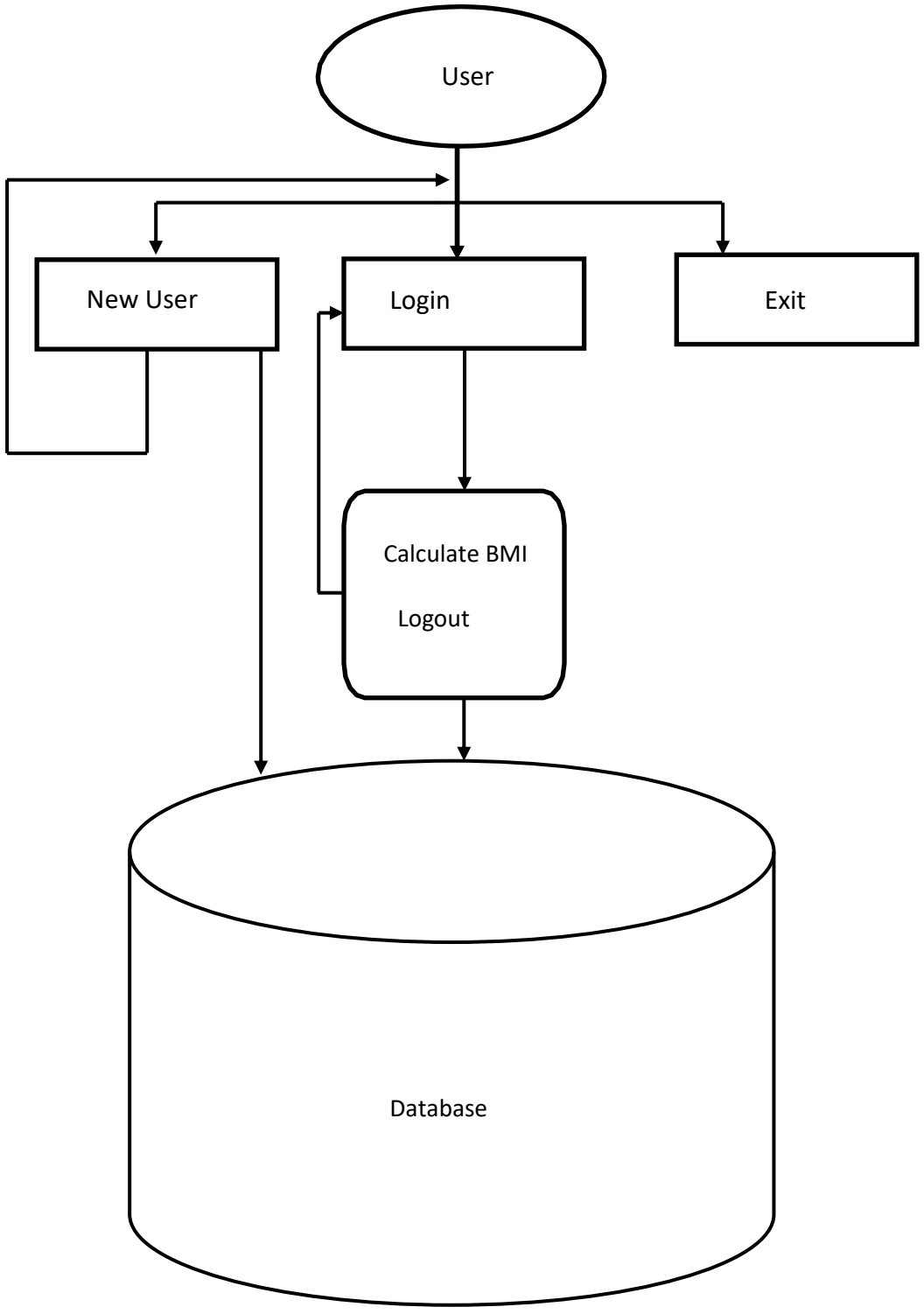
### 1.4.2 Benefits of Report Projects:

Report projects act as containers for report definitions and resources. Use projects to:

1. Organize reports and related items in one container.
2. Test report solution that include report and related items locally.
3. Preserve a set of master copies for reports and related items. after deployment, published reports can be accidentally

## 1.5 DFD/ERD/OOAD

- **E-R diagram for BMI Calculator**



# Chapter 4: TESTING AND IMPLEMENTATION

## 1.1 Testing Methodology

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same the user is being trained or later. Implementation simply means convening a new system design into operation which is the process of converting a new revised system design into an operational one.

Implementation is the realization, application, or execution of a plan, idea, model, design, specification, standard algorithm, or policy.

The design must be translated in to machine-readable form. The code generation step performs their tasks. If design is performed in a detailed manner, code generation can be accomplished mechanistically.

VISUAL BASIC is a high-level programming language which evolved from the earlierDOS version called BASIC. BASIC means Beginners' All-purpose Symbolic InstructionCode. It is a very easy programming language to learn. The code looks a lot like English Language. Different software companies produced different versions of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC

, IBM BASICA and so on. However, people prefer to use Microsoft Visual Basic today, as it is a well-developed programming language and supporting resources are available everywhere. Now, there are many versions of VB exist in the market, the most popular one and still widely used by many VB programmers is none other than Visual Basic 6. We also have VB.net, VB2005 and the latest VB2008, which is a fully object-oriented programming (OOP) language. It is more powerful than VB6 but looks more complicated to master.

VISUAL BASIC is a VISUAL and events driven Programming Language. These are the main divergence from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VB, programming is done in a graphical environment. In the old BASIC, you have to write program code for each graphical object you wish to display it on screen, including its position andits color. However, In VB, you just need to drag and drop any graphical object anywhere on the form, and you can change its color any time using the properties windows. On the other hand, because the user may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions (events). Therefore, a VB Program is made up of many subprograms, each has its own program code, and each can be executed independently and at the same time each can be linked together in one way or another.

we can create any program depending on your objective. For example, if you are a college or university lecturer, you can create educational programs to teach business, economics, engineering, computer science, accountancy, financial management, information system and more to make teaching more effective and interesting. If you are in business, you can also create business programs such as inventory management system, point-of- sale system, payroll system, financial program as well as accounting program to help manage your business and increase productivity. For those of you who like games and working as games programmer, you can create those programs as well. Indeed, there is no limit to what program you can create! There are many such programs in this tutorial, so

you must spend more time on the tutorial in order to learn how to create those programs.

Visual Basic (VB) is the third-generation event-driven programming language andintegrated development environment (IDE) from Microsoft for its COM programming model. Visual Basic is relatively easy to learn and use. Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data AccessObjects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects. Scripting languages such as VBA and VBScript are syntactically similar to Visual Basic, but perform differently.

A programmer can put together an application using the components provided with Visual Basic itself. Programs written in Visual Basic can also use the Windows API, but doing so requires external function declarations. The final release was version 6 in 1998. Microsoft's extended support ended in March 2008 and the designated successor was Visual Basic .NET (now known simply asVisual Basic).

## 1.2 Unit testing

Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements. Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.

Unit tests are automated and are run each time the code is changed to ensure that new code does not break existing functionality. Unit tests are designed to validate

the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system. This allows developers to quickly identify and fix any issues early in the development process, improving the overall quality of the software and reducing the time required for later testing.

**Objective of Unit Testing:**

The objective of Unit Testing is:

1. To isolate a section of code.
2. To verify the correctness of the code.
3. To test every function and procedure.
4. To fix bugs early in the development cycle and to save costs.
5. To help the developers to understand the code base and enable them to make changes quickly.
6. To help with code reuse

## 1.3 Module testing

Module testing is a type of software testing where individual units or components of the software are tested. The purpose of module testing is to isolate a section of code and verify its correctness. Module testing is usually performed by the development team during the early stages of software development. However, it can also be done by independent testers as part of regression testing. There are various methods of module testing, but the most common one is black-box testing. In black-box testing, the test cases are designed based on the functionality of the code, without taking into consideration its internal structure. Other methods include white-box testing (which looks at the internal structure), grey- box testing (which combines both black-box and white-box testing), and glass-box testing (which tests all possible inputs and outputs). No matter which method youchoose, module testing is an important part of ensuring the quality of your

software. In this blog post, we will take a closer look at module testing and how it can benefit your software development process.

**Objectives of Module Testing**

- There are several objectives of module testing:

  - Module working as expected: To ensure that each module is working as expected.
  - Interfaces working properly: To ensure that the interfaces between modules are working as expected.
  - Overall system working properly: To ensure that the overall system functions as expected.
  - Detect errors introduced during development: To find errors that may have been introduced during the development process.
  - Increase confidence in software: To provide confidence that the system is ready for release.

**Inputs of Module Testing**

- Inputs to module testing include the following:

    - Requirements: The first input to module testing is the requirements. The tester must have a clear understanding of what the module is supposed to do. This understanding is typically captured in the form of requirements, which can be in the form of user stories, use cases, or functional specifications.
    - Design: The second input to module testing is the design. The design provides the tester with a high-level view of how the module is implemented. The design should include a description of the interfaces between the module and the rest of the system.
    - Code: The third input to module testing is the code. The code is the actual implementation of the module. The tester will use the code to execute tests and verify the results.
    - Test Cases: The fourth input to module testing is the test cases. The test cases define the specific tests that will be executed against the module. The test cases should be designed to exercise all the functionality of the module.
    - Test Data: The fifth input to module testing is the test data. The test data is used to execute the test cases. The test data should be designed to exercise all the different inputs to the module.
    - Test Environment: The sixth input to module testing is the test environment. The test environment is the environment in which the tests will be executed.
    - Test Tools: The seventh input to module testing is the test tools. The test tools are the software and hardware that will be used to execute the tests. The test tools should be chosen so that they are appropriatefor the type of testing that is being done.

## 1.4 System testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The resultof system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of eithersystem requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS). System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing. System Testing is a black-box testing. System Testing is performed after the integration testing and before the acceptance testing.

Design is the first step into the development phase for any engineered productor system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used.

The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has

undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user- oriented document to a document to the programmers or database personnel.

## 1.5 Alpha/ Beta Testing

**Alpha Testing** is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the **user acceptance tastings**. This is referred to as alpha testing only because it is done early on, near the end of the development of the software. Alpha testing is commonly performed by homestead software engineers or quality assurance staff. It is the last testing stage before the software is released into the real world.

1. Alpha testing is a software testing stage that takes place early in the development process, typically after the code has been written and before the final product is released to the public. Alpha testing is performed by a select group of internal stakeholders, such as developers, testers, and members of the product team.
2. The purpose of alpha testing is to identify and resolve critical bugs and issues in the software before it is released to the public. Alpha testing is performed in a controlled environment, such as a lab or a test network, and is used to simulate real-world use cases and identify any potential problems.
3. During alpha testing, the software is evaluated against a set of predetermined acceptance criteria and is tested for functionality, usability, performance, and stability. Alpha testing provides an opportunity to identify and fix bugs and issues before they reach end-users, ensuring that the final product is of high quality and meets the needs of the target audience.

**Advantages of alpha testing include:**

1. Early identification of bugs and issues: Alpha testing allows for the early identification of bugs and issues, providing an opportunity to fix them before they reach end-users.
2. Improved quality: By identifying and fixing bugs and issues early in the development process, alpha testing helps to improve the overall quality of the software.
3. Increased user satisfaction: Alpha testing helps to ensure that the software meets the needs of the target audience, leading to increased user satisfaction.
4. Faster resolution of problems: Alpha testing allows for the rapid resolution of problems, reducing the likelihood of further issues down the line.
5. Cost savings: By identifying and fixing issues early in the development process, alpha testing can help to save time and money by avoiding the need for more extensive testing and bug fixing later on.

**Objective of Alpha Testing:**

1. The objective of alpha testing is to refine the software product by finding the bugs that were not discovered during the previous tests.
2. The objective of alpha testing is to refine the software product by fixing the bugs that were not discovered during the previous tests.
3. The objective of alpha testing is to involve customers deep into the process of development.
4. The objective of alpha testing is to give better insight into the software's reliability at the early stages of development.
5. The main objective of alpha testing is to identify and resolve critical bugs and issues in the software before it is released to the public. The goal is to assess the software's overall quality, functionality, usability, performance, and stability in a controlled environment, and to ensure that it meets the needs and expectations of the target audience.
6. During alpha testing, the software is evaluated against a set of predetermined acceptance criteria, and any issues or bugs that are identified are documented and reported back to the development teamfor resolution. The objective of alpha testing is to provide an early opportunity to identify and fix bugs and issues, reducing the likelihood of them affecting end-users and potentially causing damage to the software's reputation.
7. Overall, the objective of alpha testing is to improve the quality of the software, ensure that it meets the needs of the target audience, and reduce the risk of issues and bugs affecting end-users after the software has been released.

**Beta Testing** is performed by real users of the software application in a real environment. Beta testing is one of the types of **User Acceptance Testing**. A Beta version of the software, whose feedback is needed, is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing helps in minimization of product failure risks and it provides increased quality of the product through customer validation. It is the last test before shipping a product to the customers. One of the major advantages of betatesting is direct feedback from customers.

Beta testers are "real" users and conduct their testing in a production environment running on the same hardware, networks, etc., as the final release. This also means it's the first chance for full security and reliability testing because those tests can't be conducted in a lab or stage environment.

Beta tests can either be open or closed. In an open test, anyone can use the product and is usually presented with some messaging that the product is in betaand given a method for submitting feedback. In closed beta, the testing is limited to a specific set of testers, which may be composed of current customers, early adopters, and/or paid beta testers. Sometimes they are conducted by diverting a certain percentage of users to the beta site instead of the current release.

**Some features of Beta testing**

A pool of potential end-users performs it to give unbiased feedback on the product.
It is performed at the end user's site.
The beta test execution cycle is generally smaller than the in-house (alpha) testing cycle.
Since the end-users testing, the beta version doesn't have access to the source code of the product, so it involves black box testing only.

**Characteristics of Beta Testing:**

1. Beta Testing is performed by clients or users who are not employees of the company.
2. Reliability, security, and robustness are checked during beta testing.
3. Beta Testing commonly uses black-box testing.
4. Beta testing is carried out in the user's location.
5. Beta testing doesn't require a lab or testing environment.

## 1.6 White box testing

The term 'white box' is used because of the internal perspective of the system. The clear box or white box, or transparent box name denotes the ability to see through the software's outer shell into its inner workings.

It is performed by Developers, and then the software will be sent to the testing team, where they perform black-box testing. The main objective of white-box testing is to test the application's infrastructure. It is done at lower levels, as it includes unit testing and integration testing. It requires programming knowledge, as it majorly focuses on code structure, paths, conditions, and branches of a program or software. The primary goal of white-box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

its techniques analyze the internal structures the used data structures, internal design, code structure, and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear boxtesting or structural testing. White Box Testing is also known as transparent testing or open box testing.

White box testing is a software testing technique that involves testing the internal structure and workings of a software application. The tester has access to the source code and uses this knowledge to design test cases that can verify the correctness of the software at the code level.

White box testing is also known as structural testing or code-based testing, and it is used to test the software's internal logic, flow, and structure. The tester createstest cases to examine the code paths and logic flows to ensure they meet the specified requirements.

**Black box testing**

The primary source of black-box testing is a specification of requirements that are stated by the customer. It is another type of manual testing. It is a software testing technique that examines the functionality of the software without knowing its internal structure or coding. It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function. In this testing, the test engineer analyzes the software against requirements, identifies the defects or bugs, and sends it back to the development team. In this method, the tester selects a function and gives input value to examine its functionality, and checks whether the function is giving the expected output or not. If the function produces the correct output, then it is passed in testing, otherwise failed.

Black box testing is less exhaustive than White Box and Grey Box testing methods. It is the least time-consuming process among all the testing processes. The main objective of implementing black box testing is to specify the business needs or the customer's requirements.

**Black box testing can be done in the following ways:**

1. **Syntax-Driven Testing –** This type of testing is applied to systems that canbe syntactically represented by some language. For example-compilers, language that can be represented by context-free grammar. In this, the test cases are generated so that each grammar rule is used at least once.

- The methodology of syntax testing can be perceived and understood through following steps in the sequential order:
- The very first step of the syntax testing involves the identification of the target language or format.
- Thereafter, syntax of the language is defined, as specified in the formal notation. As each and every input has some syntax, which may be formally pre-specified, undocumented, etc.

- The final step involves testing and debugging the syntax to ensure its completeness & consistency. Generally, the syntax is tested using two conditions as stated below.
    - Testing the normal condition using the covering set of paths of the syntax graph, for the minimum necessary requirements.
    - Testing the garbage condition*, using invalid set of input data.

**2. Equivalence partitioning –** It is often seen that many types of inputs work similarly so instead of giving all of them separately we can group them and test only one input of each group. The idea is to partition the input domain of the system into several equivalence classes such that each member of the class works similarly, i.e., if a test case in one class results in some error, other members of the class would also result in the same error.

**Guidelines for Equivalence Partitioning:**

- If the range condition is given as an input, then one valid and two invalid equivalence classes are defined.
- If a specific value is given as input, then one valid and two invalidequivalence classes are defined.
- If a member of set is given as an input, then one valid and one invalid equivalence class is defined.
- If Boolean no. is given as an input condition, then one valid and one invalid equivalence class is defined.

## 1.7 Source code

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import java.sql.*;
public class SQLDatabaseConnection {
// Connect to your database.
static final String connectionUrl = "jdbc:mysql://localhost:3306/bmi_records_db";
static final String user = "root";
static final String spassword = "12345678";
private static String Name;
private static String Prn;
private static String Course;
private static String Gmailid;
private static String Phone;
private static int Age;
private static String userid;
private static String password;
private static int weight;
private static double height;
private static double BMI;
private static double BMI  Prime;
private static String newUserId;
private static String newPassword;
// User ID and Password map
private static Map<String, String> users = new HashMap<>();
private static String loggedInUser = null;
public SQLDatabaseConnection(double height, int weight)
{
this.height = height;
this.weight = weight;
}
public double calculateBMI() {
return weight / (height * height);
}
public double calculateBMIPrime(double BMI)
{
return BMI / 25;
}
public void displayResults(double BMI, double BMI_Prime)
{
System.out.println("------------User Details--------------------");
System.out.println("User Name  :" + Name);
System.err.println("Prn Number :" + Prn);
System.out.println("Course Name:" + Course);
System.out.println("Email ID   :" + Gmailid);
```

```
System.out.println("Phone No   :" + Phone);
System.out.println("Age        :" + Age);
System.out.println();
System.err.println("----------BMI Calculation Results----------");
System.out.println("Healthy BMI range: [18.5 kg/m2 - 25 kg/m2]\n");
System.out.println("Your Body Mass Index is  : " + BMI);
if (BMI < 18.5)
{
System.out.println("You are Underweight ");
}
else if (BMI < 25)
{
System.out.println("You are within a healthy weight range");
}
else if (BMI < 30)
{
System.out.println("You are Overweight");
} else if (BMI < 35)
{
System.out.println("You are Obese Class I ");
} else if (BMI < 40)
{
System.out.println("You are Obese Class II ");
} else
{
System.out.println("You are Obese Class III ");
}
System.out.println("Your BMI prime  : " + BMI_Prime);
System.out.println("Please Wait...");
Try
{
Thread.sleep(9000); // Sleep for 3 seconds
} catch (InterruptedException e)
{
e.printStackTrace();
}
 }
public static String generateRandomPassword() {
String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
StringBuilder sb = new StringBuilder();
for (int i = 0; i < 8; i++)
{
int index = (int) (Math.random() * characters.length());
sb.append(characters.charAt(index));
}
return sb.toString();
}
public static void clearScreen() {
System.out.print("\033[H\033[2J");
```

```
System.out.flush();
}
public static void main(String[] args) throws Exception {
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con = DriverManager.getConnection(connectionUrl, user, spassword);

Scanner input = new Scanner(System.in);
SQLDatabaseConnection calculator = null;

while (true)
{
clearScreen();
if (loggedInUser == null)
{
System.out.println("------------Welcome to BMI Calculator!------------");
System.out.println("1. New User Registration");
System.out.println("2. Login");
System.out.println("3. Exit");
System.out.print("Enter your choice: ");
int option = input.nextInt();

switch (option)
{

case 1:
clearScreen();
System.out.println("New User Registration");

System.out.print("Enter Name: ");
Name = input.next();System.out.print("Enter Prn:");
Prn = input.next();
System.out.print("Enter Course Name:");
Course = input.next();
System.out.print("Enter Email ID: ");
Gmailid = input.next();
System.out.print("Enter Phone Number: ");
 Phone = input.next();
System.out.print("Enter Age: ");
Age = input.nextInt();

// Generate user ID
newUserId = "user" + ((users.size() + 1));

// Generate random password
newPassword = generateRandomPassword();
 users.put(newUserId, newPassword);

clearScreen();
 System.out.println("User registration successful!");
```

```
 System.out.println("Please Wait...");
Try
{
Thread.sleep(3000); // Sleep for 3 seconds
}
 catch (InterruptedException e)
{
 e.printStackTrace();
}
 clearScreen();

 System.out.println("Please Collect Your UserId and Password!... ");

try
{
 Thread.sleep(3000); // Sleep for 3 seconds
}
catch (InterruptedException e)
{
 e.printStackTrace();
}
clearScreen();
System.out.println("User ID: " + newUserId);
System.out.println("Password: " + newPassword);
System.out.println("Please Wait...");
userid = newUserId;
password = newPassword;

try
{
 Thread.sleep(4000); // Sleep for 3 seconds
}
catch (InterruptedException e)
{
 e.printStackTrace();
}
break;

case 2:
clearScreen();
 System.out.println("User ID: " + userid);
System.out.println("Password: " + password);
System.out.println("-------------User Login-----------------");
System.out.print("Enter User ID: ");
String userId = input.next();
System.out.print("Enter Password: ");
String password = input.next();
clearScreen();
 if (users.containsKey(userId) && users.get(userId).equals(password)) {
```

```
 System.out.println("Login successfully!");
System.out.println("Please Wait...");
 loggedInUser = userId;
 try
{
Thread.sleep(3000); // Sleep for 3 seconds
}
 catch (InterruptedException e)
{
 e.printStackTrace();
}
}
Else
{
 clearScreen();
 System.out.println("Invalid User ID or Password!");
 System.out.println("Please Wait...");

 Try
{
 Thread.sleep(3000); // Sleep for 3 seconds
}
catch (InterruptedException e)
{
e.printStackTrace();
}
}
break;
case 3:
clearScreen();
System.out.println("Exiting BMI Calculator...");
input.close();

try
{
Thread.sleep(3000); // Sleep for 3 seconds
}
catch (InterruptedException e)
{
 e.printStackTrace();
 }
clearScreen();
System.out.println("Thanks a lot For Using My Program...");

try
{
 Thread.sleep(3000); // Sleep for 3 seconds
}
catch (InterruptedException e)
```

```
{
 e.printStackTrace();
 }
 clearScreen();

System.exit(0);
break;
default:
System.out.println("Invalid Option... ");

}
}
else
{
clearScreen();

 System.out.println("Welcome back, " + loggedInUser + "!");
 System.out.println("1. Calculate BMI");
 System.out.println("2. Logout");
 System.out.print("Enter your choice: ");
 int option = input.nextInt();

 switch (option) {
 case 1:
 clearScreen();
 System.out.print("Please Enter Weight in Kg : ");
 weight = input.nextInt();

System.out.print("Please Enter Height in Meters: ");
height = input.nextDouble();

clearScreen();

calculator = new SQLDatabaseConnection(height, weight);
BMI = calculator.calculateBMI();

BMI  Prime = calculator.calculateBMIPrime(BMI);

calculator.displayResults(BMI, BMI  Prime);
 PreparedStatement sp = con.prepareStatement("insert into CalculateBmiData values('" + newUserId + "','" +
newPassword+ "','" + Name+ "','" + Prn + "','" + Course + "','" + Gmailid + "','" + Phone + "','" + Age + "','" +
weight + "','" + height + "','" + BMI + "','" + BMI  Prime + "')");

sp.executeUpdate();

try
{
 Thread.sleep(5000); // Sleep for 2 seconds
}
```

```
 catch (InterruptedException e)
{
e.printStackTrace();
 }
 break;
case 2:
clearScreen();
 System.out.println("Logging out...");
 try
{
 Thread.sleep(3000); // Sleep for 3 seconds
}
catch (InterruptedException e)
{
e.printStackTrace();
 }
loggedInUser = null;
break;

default:
 System.out.println("Invalid option... ");
 }
}
 clearScreen();

}

}

}
```

## 1.8 OUTPUT

Figure 4.1

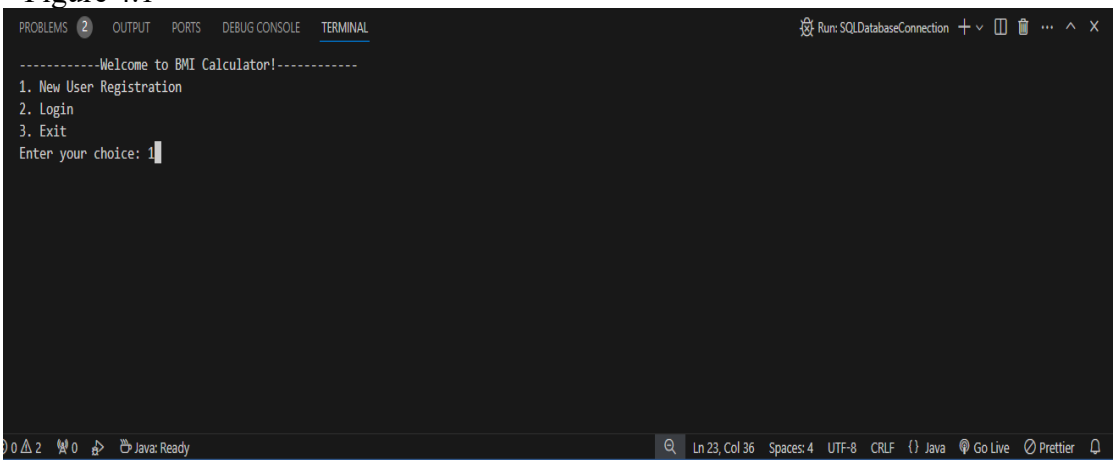



Figure 4.2 #Auto Redirect Display

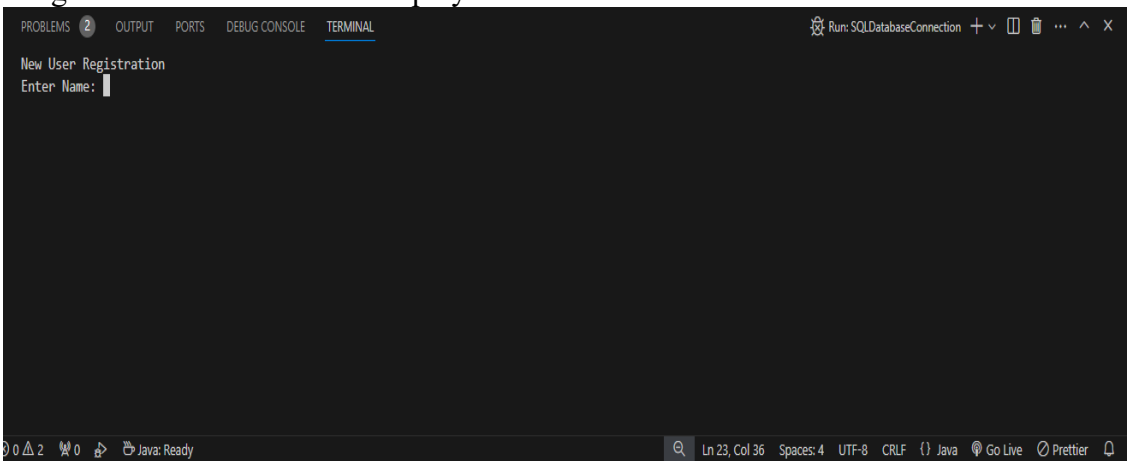



Figure 4.3

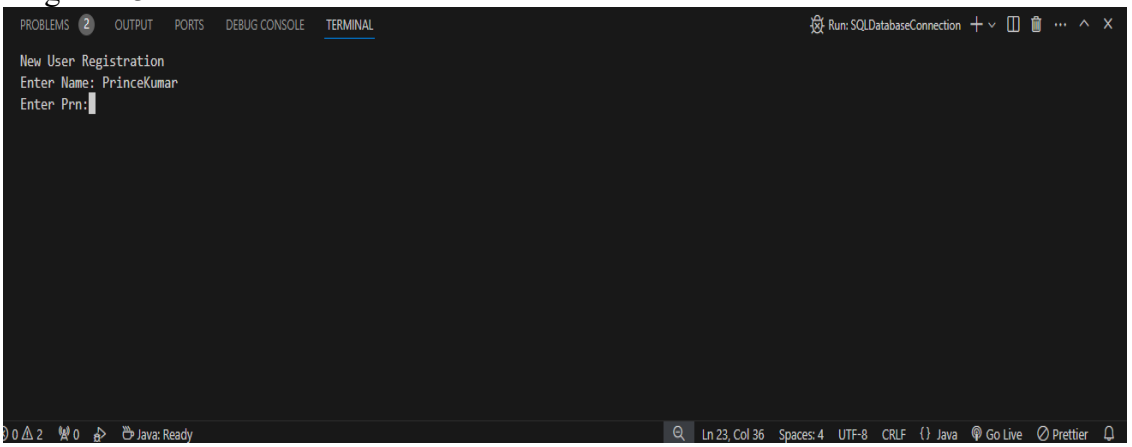


Figure 4.4

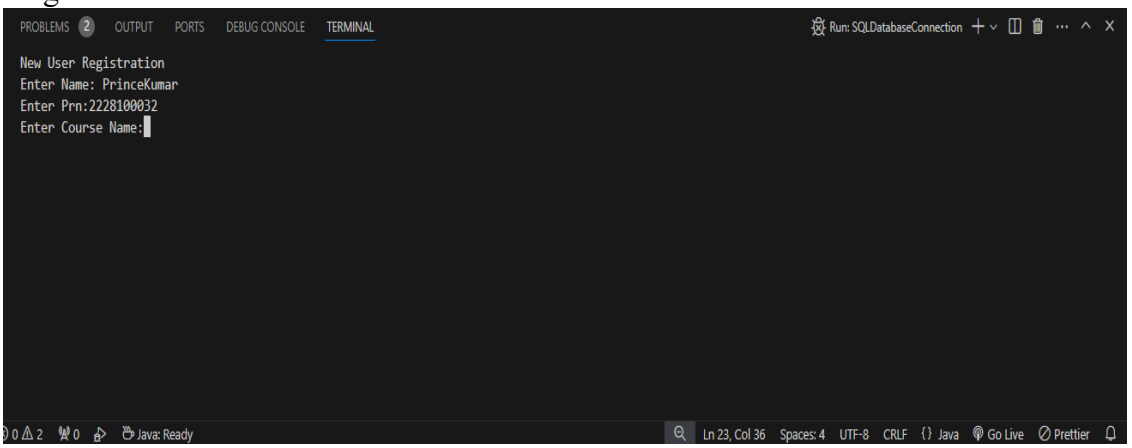



Figure 4.5

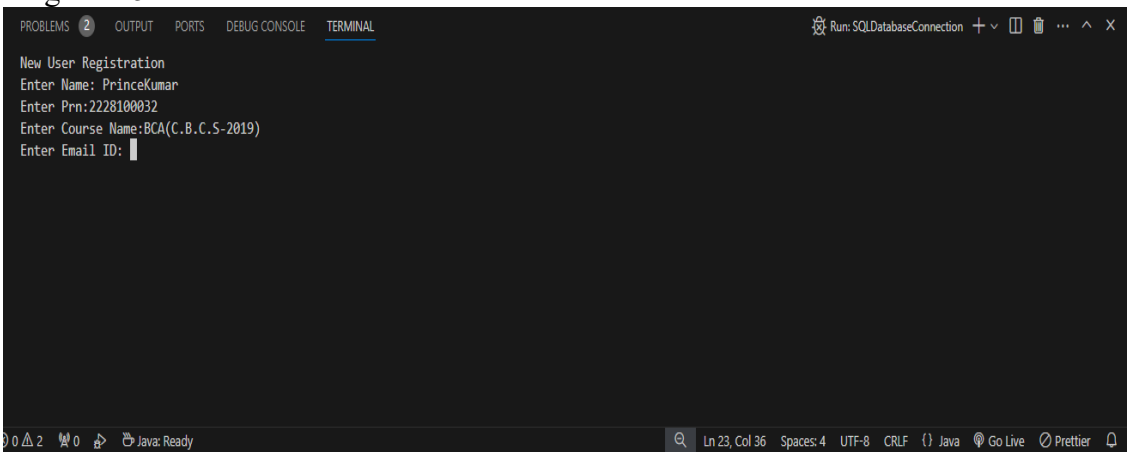



Figure 4.6

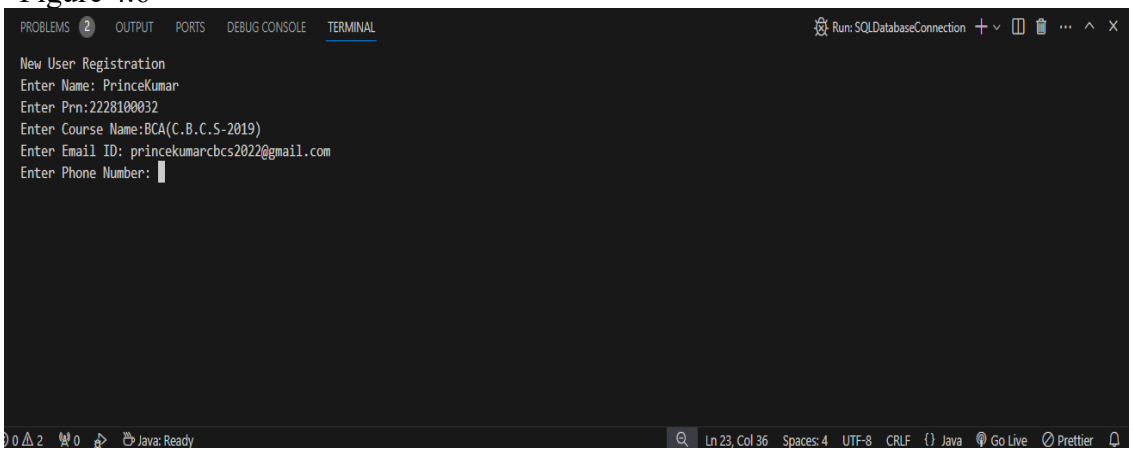


Figure 4.7

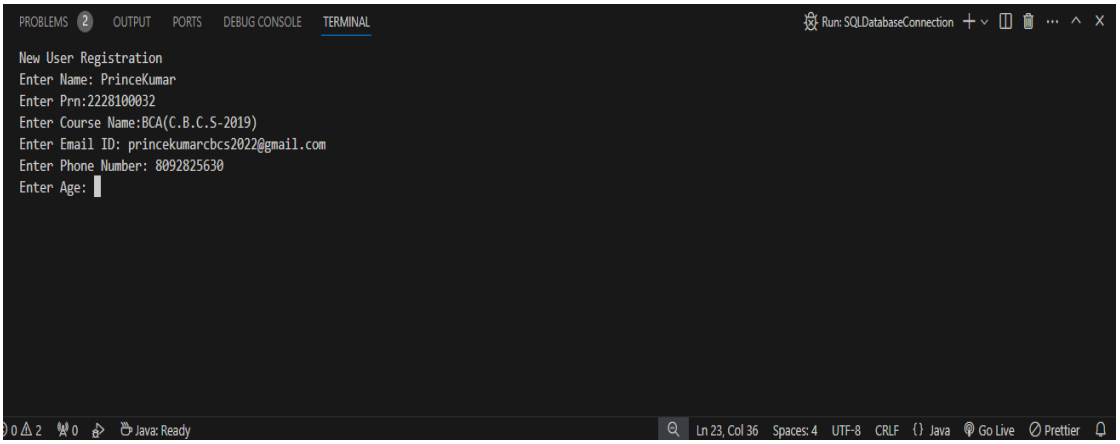



Figure 4.8

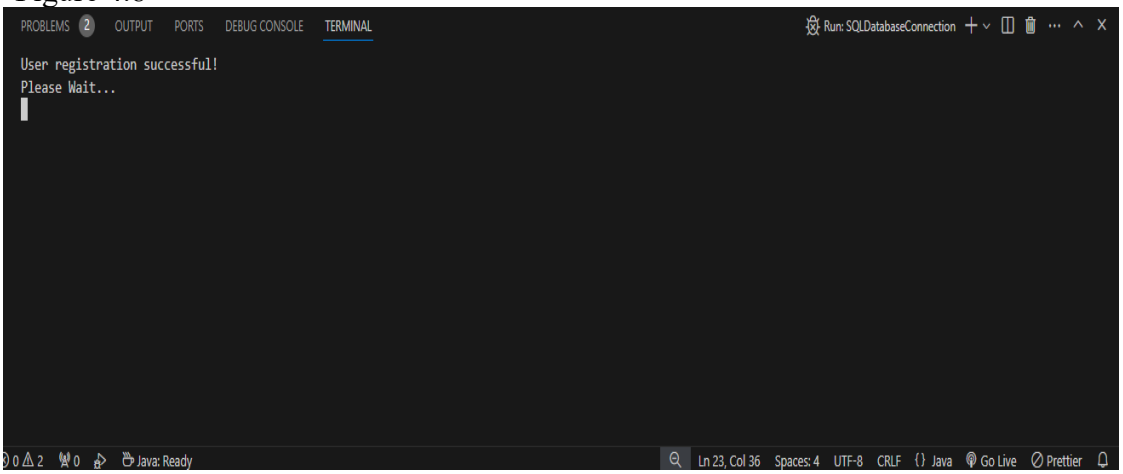



Figure 4.9 #Auto Redirect Display

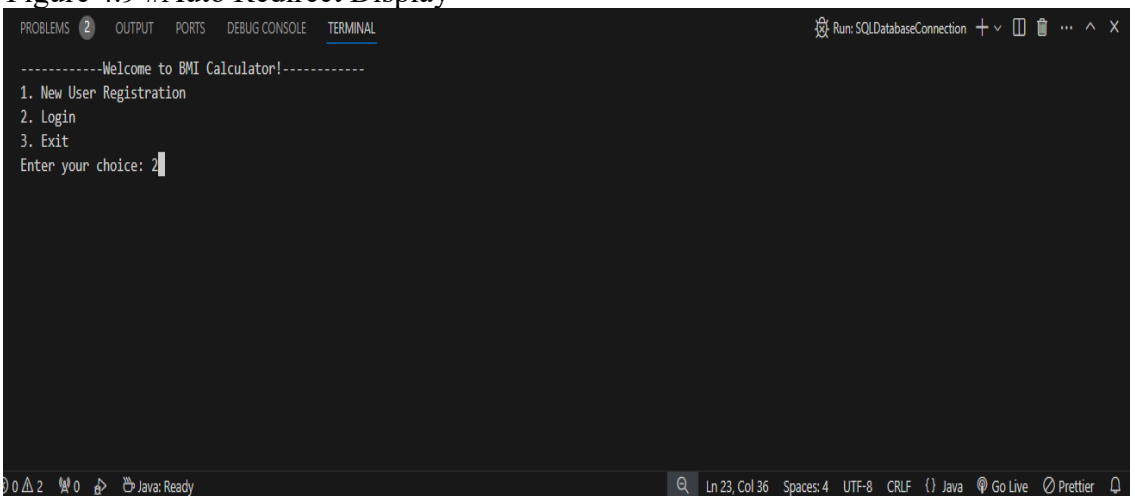


Figure 4.10

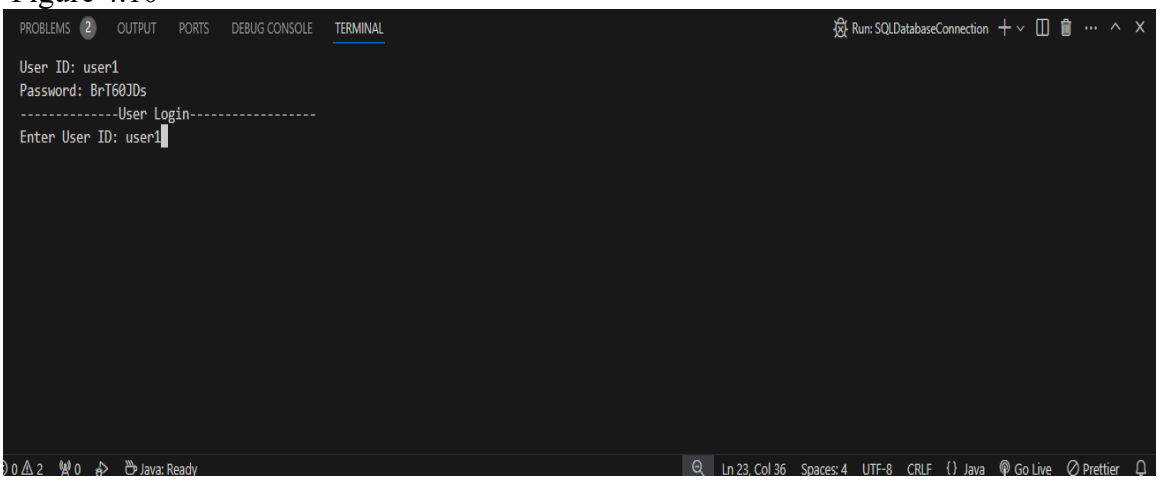



Figure 4.11

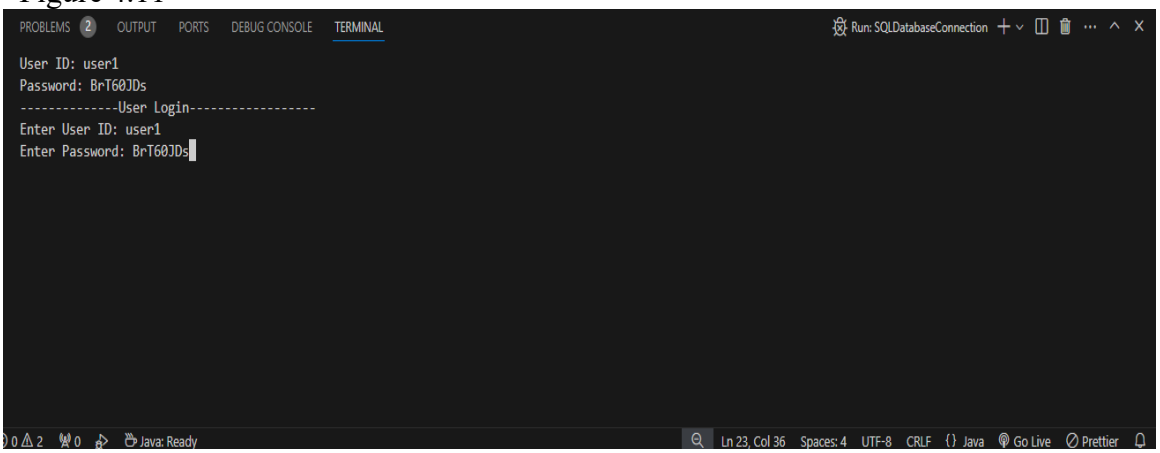



Figure 4.12

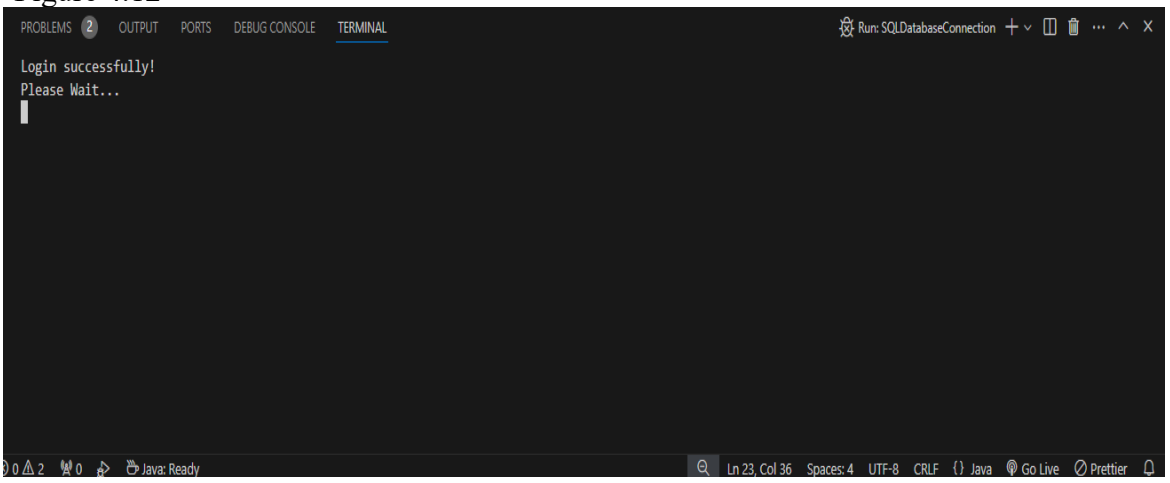


Figure 4.13 #Auto Redirect Display

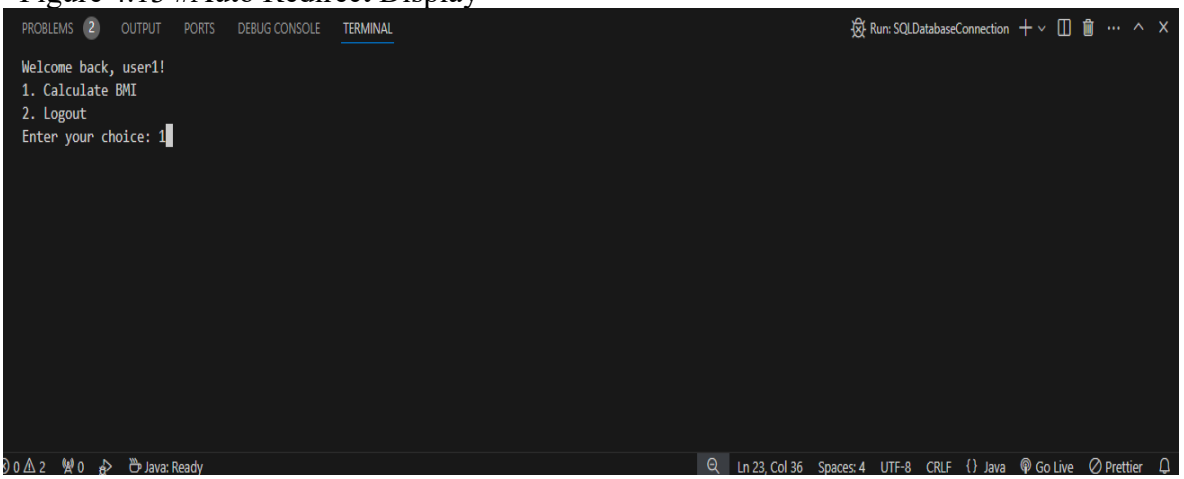



Figure 4.14

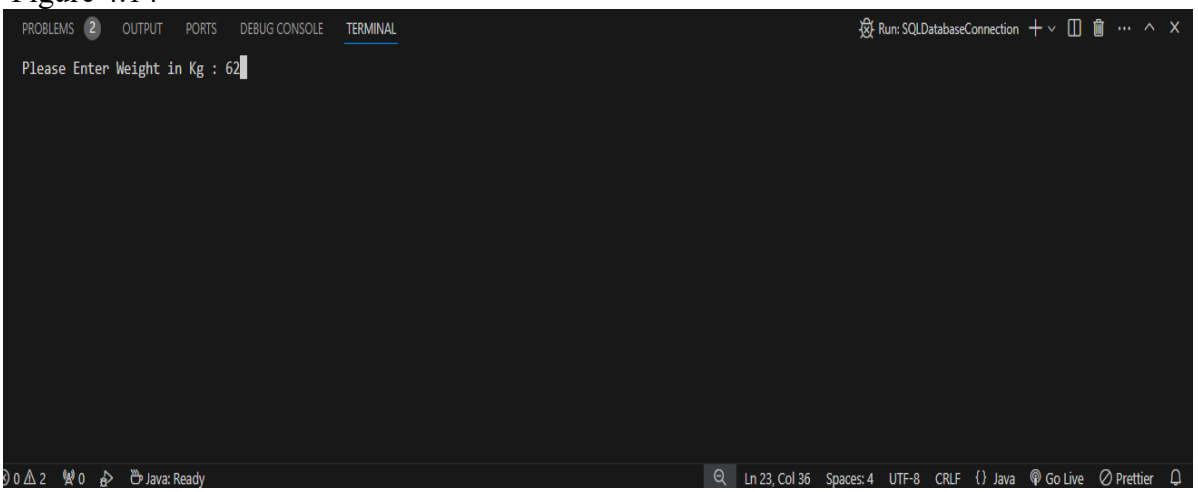



Figure 4.15

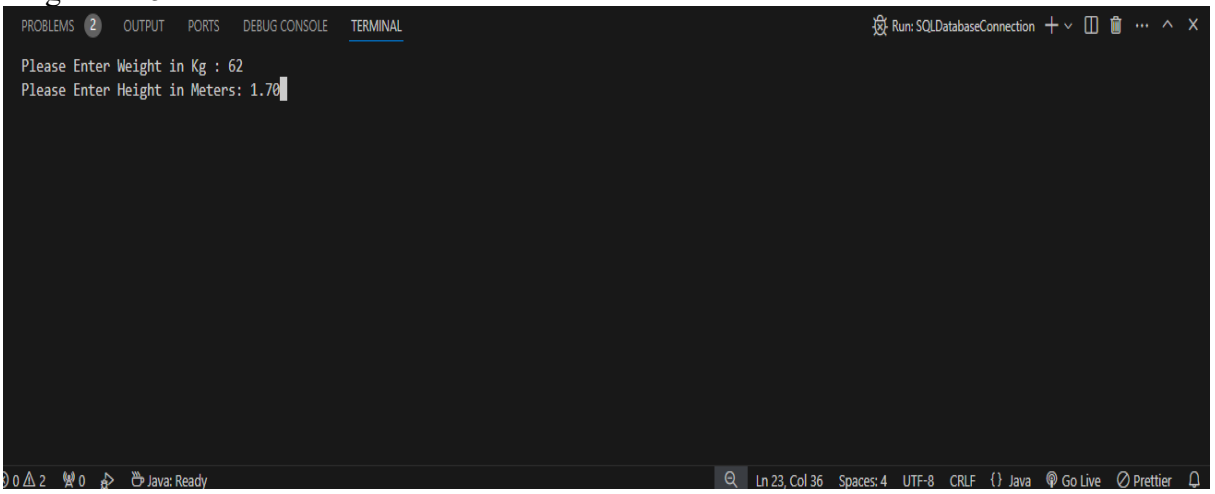


Figure 4.16 #Auto Redirect Display

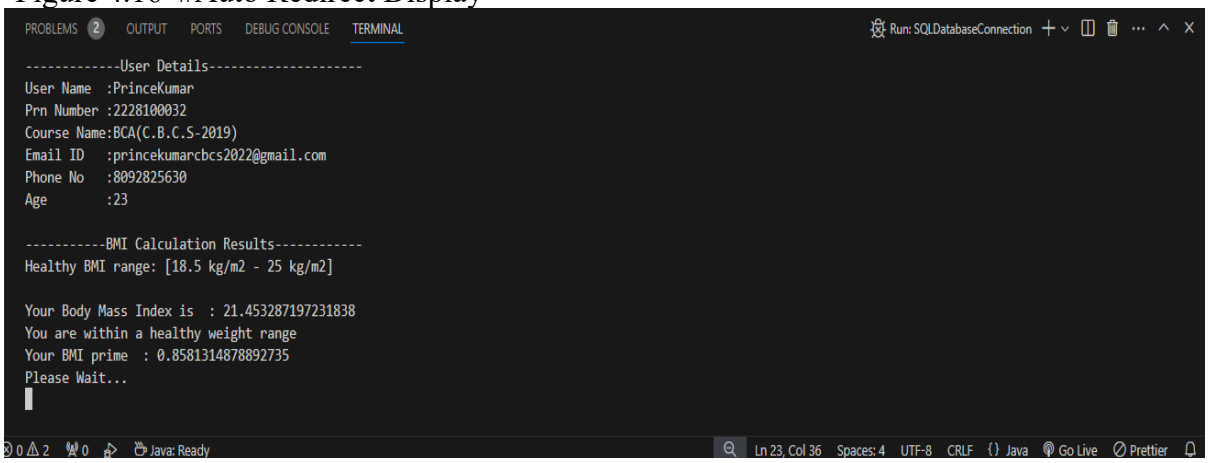



Figure 4.17 #Auto Redirect Display

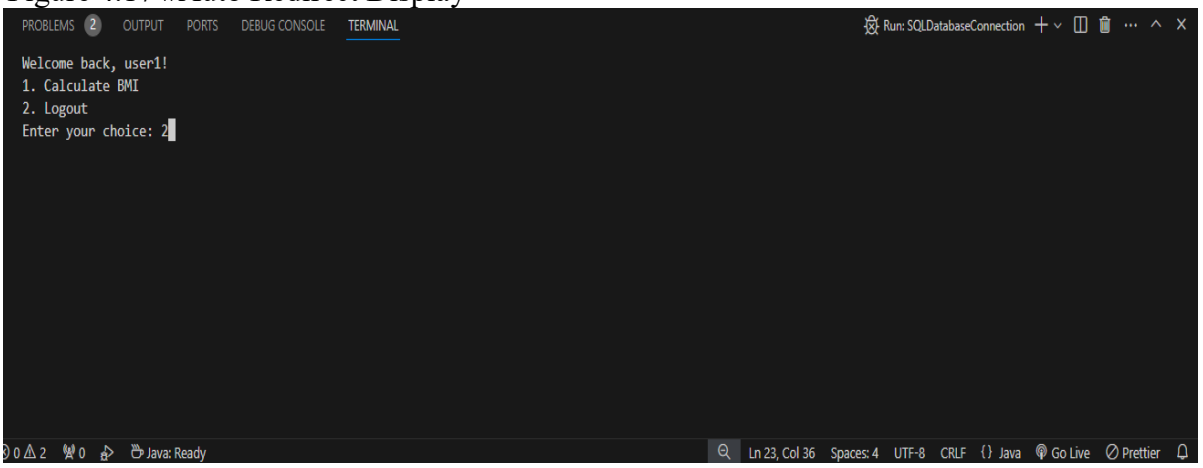



Figure 4.18

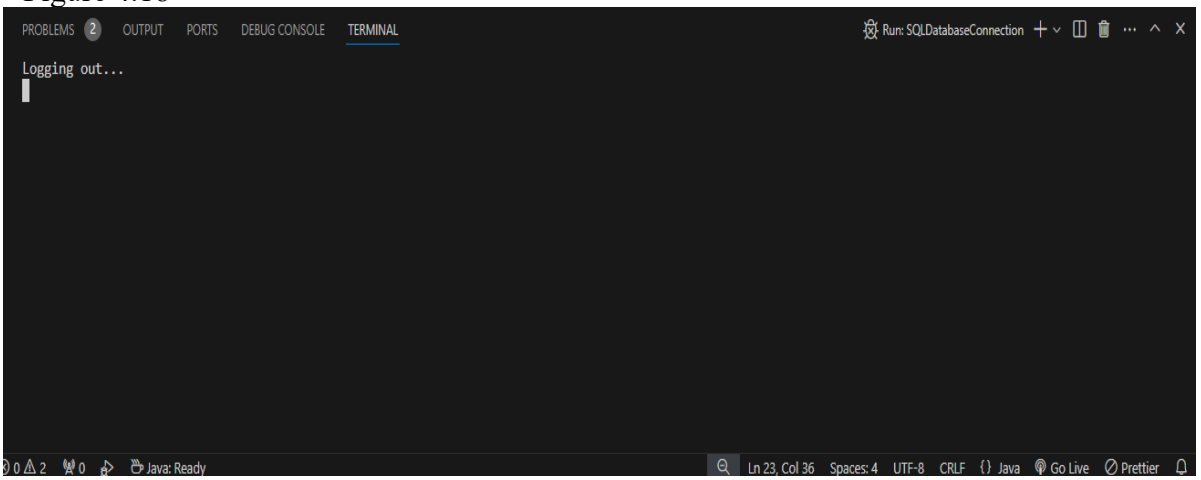


Figure 4.19 #Auto Redirect Display

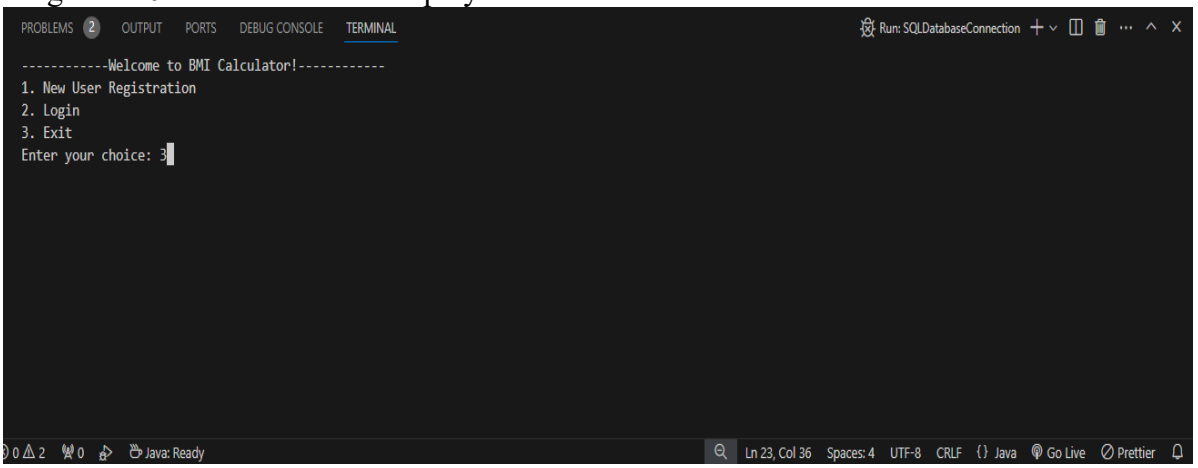



Figure 4.20

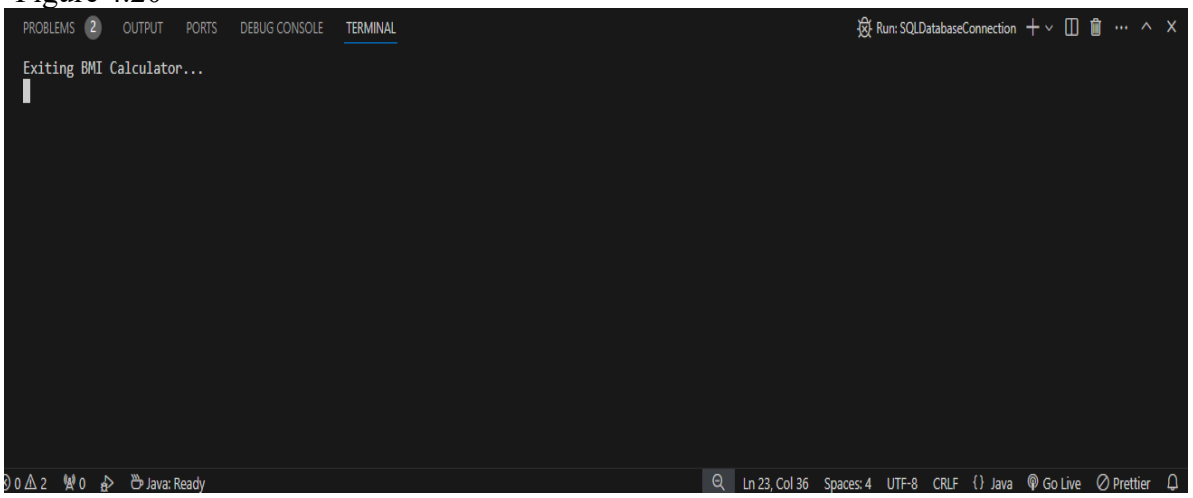



Figure 4.21 #Auto Redirect Display

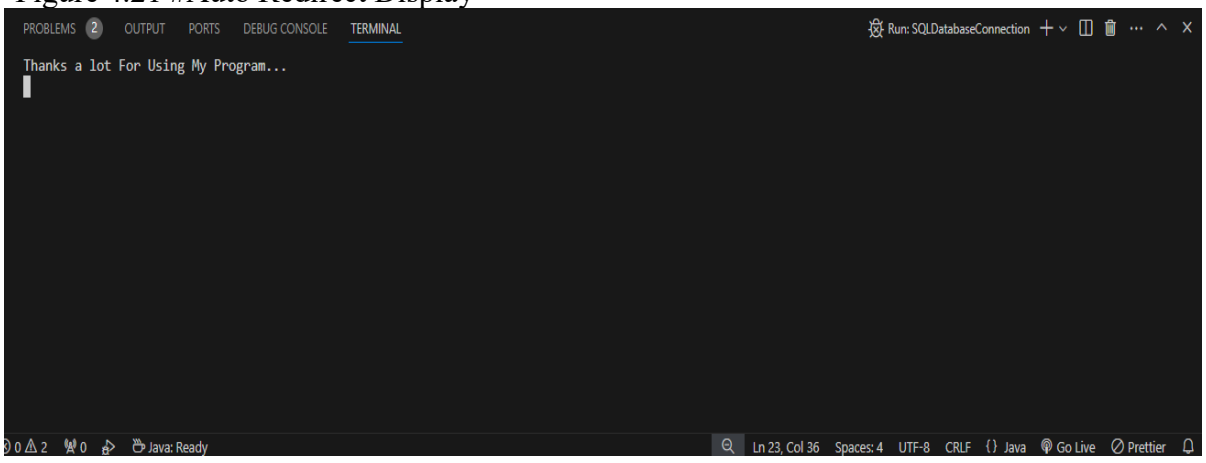
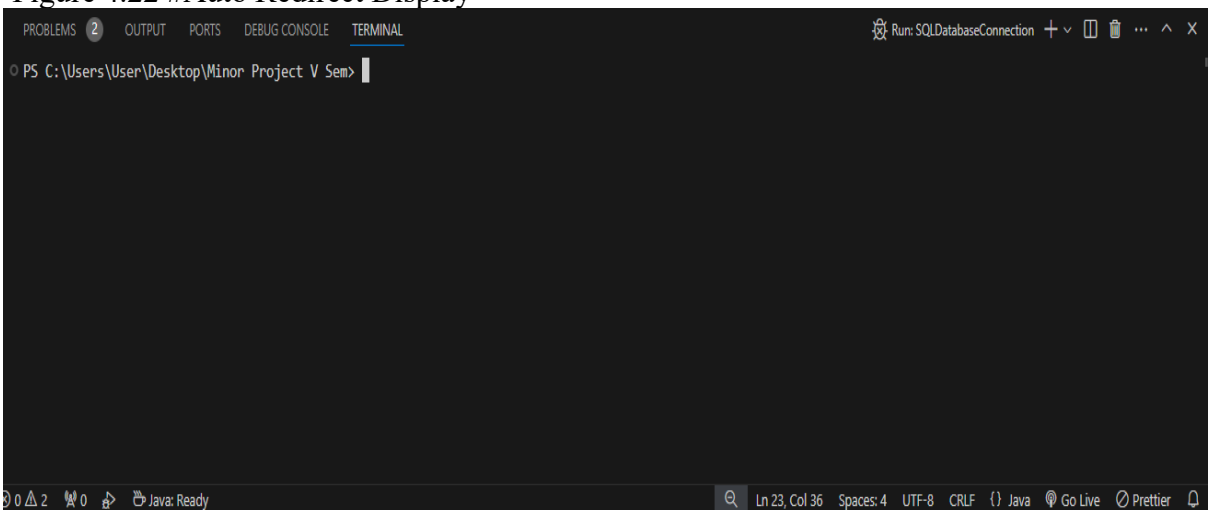


Figure 4.22 #Auto Redirect Display



## 1.9 Database

Figure 4.23 # Enter Password .

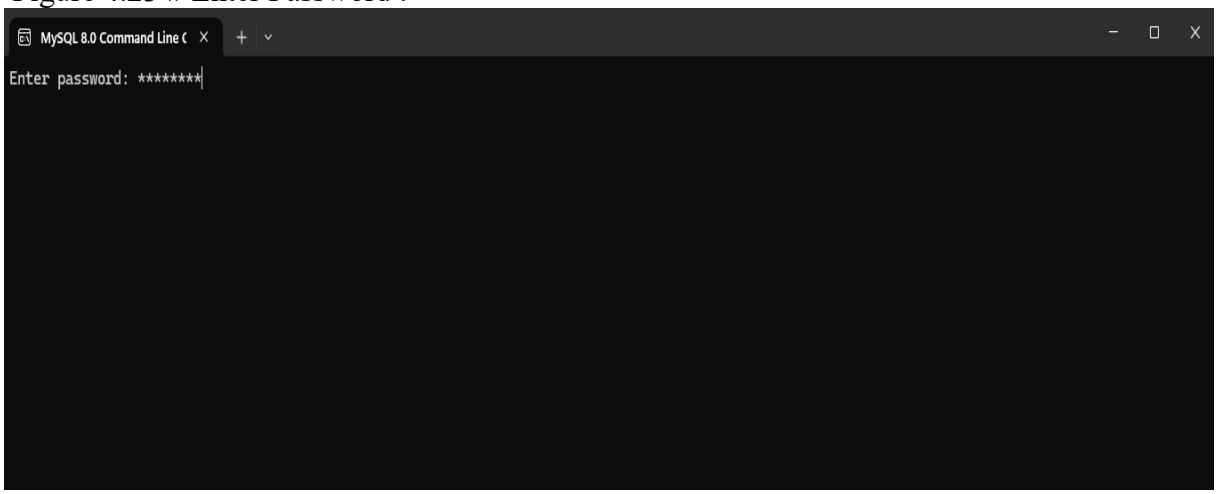



Figure 4.24

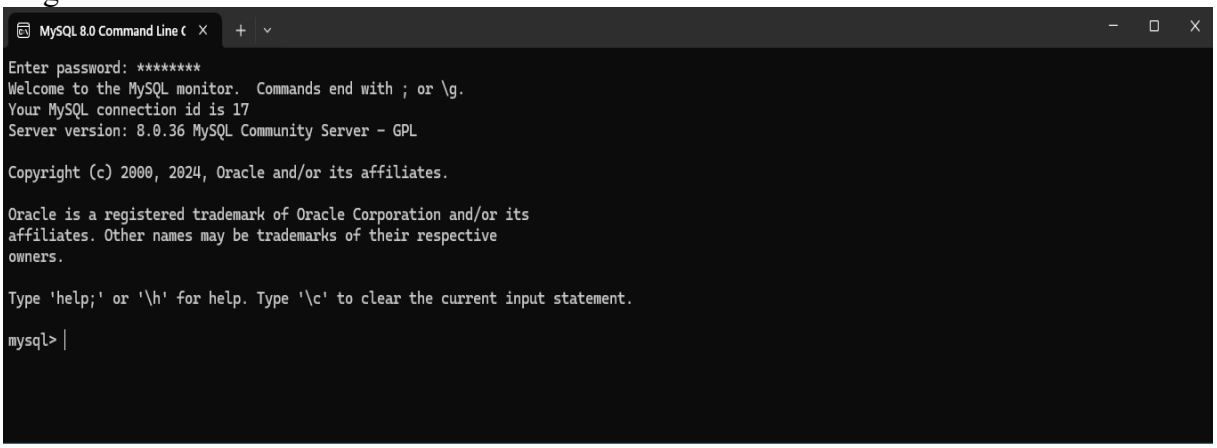



Figure 4.25 # Show Databases.

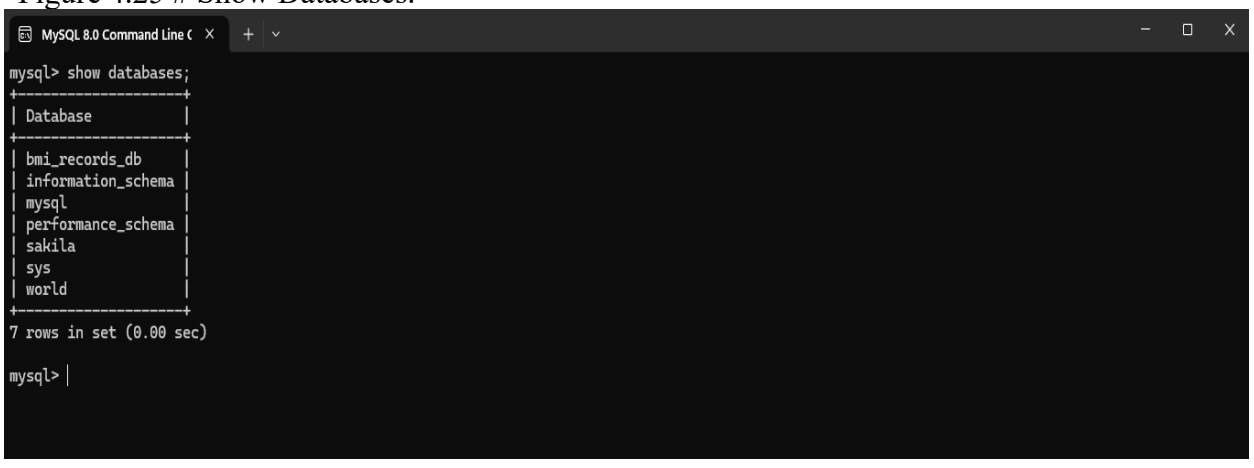
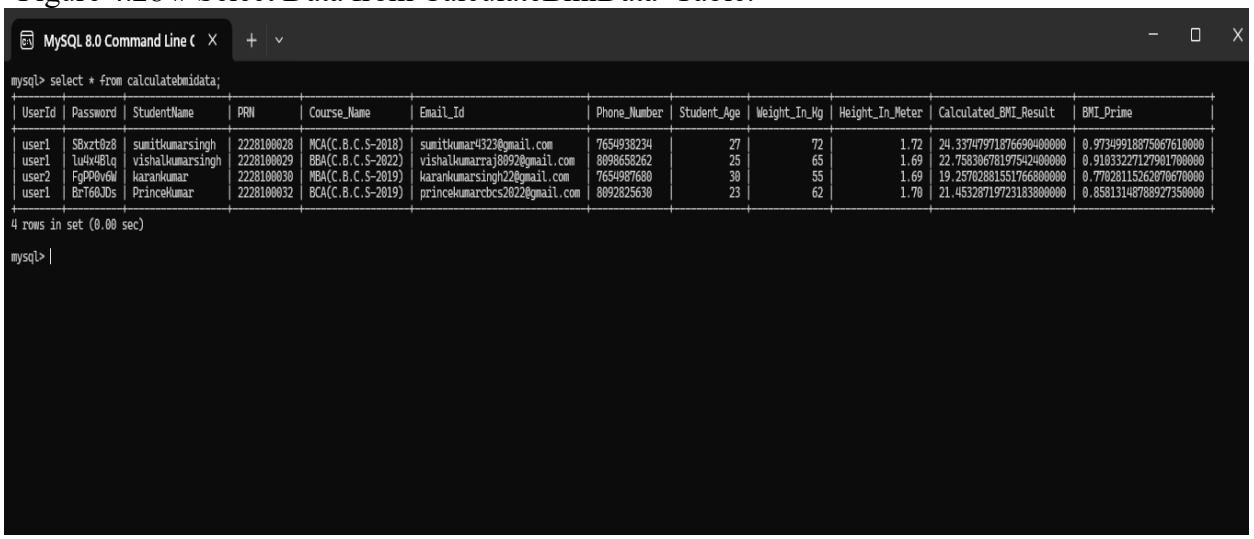



Figure 4.26 # Use Database.


Figure 4.27 # Show Tables.


Figure 4.28 # Select Data from CalculateBmiData  Table.



# Chapter 5: CONCLUTION AND REFERENCES

## 1.1 HARDWARE REQUIRMENTS

Processor : - X86 Compatible processor with 1.7 GHz
Clockspeed RAM : - 512 MB or more
Hard Disk : - 20 GB
Monitor : -
VGA/SVGAKeyboard : -
104 Keys
Mouse : - 2 buttons/ 3buttons

## 1.2 SOFTWARE REQUIRMENTS

Operating System: - Windows 2000/XP

Front end : - Visual Studio Latest
Back end : - SQL Server 2022

# Conclusion

After finishing The BMI calculator offers a straightforward method for assessing weight status and potential health risks, but it should be interpreted cautiously and supplemented with additional information for a comprehensive understanding of an individual's health profile.

The BMI calculator is a valuable tool for assessing an individual's weight status based on their height and weight. It provides a numerical value that categorizes individuals into different weight classifications, such as underweight, normal weight, overweight, or obese. This classification can serve as a useful starting point for discussions about health and wellness.

## 1.4 Bibliography

To make this project I have taken help of my subject's teachers and friends.

I used: - Notepad, Internet

**Site:** **www.studytonight.com**

**www.studytonight.com**

**www.codewithc.com**

**www.geeksforgeeks.com**

**www.codeproject.com**

**Thank you**