VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JNANA SANGAMA", BELAGAVI-590018

A DBMS LABORATORY AND MINI PROJECT (18CSL58)

REPORT ON

TIBBLES - A TIMETABLE MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Engineering

In

Artificial Intelligence & Machine Learning

By

ANUSHA PHANIRAJ

1KS20AI004

Under the guidance of

Ms. Lakshmi K K

Asst. Prof, Dept. Of AIML

Ms. Anu Mathews

Asst. Prof, Dept. Of AIML

Department of Artificial Intelligence & Machine Learning

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-56010

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Artificial Intelligence & Machine Learning

CERTIFICATE

This is to certify that Mini Project work entitled **"TIBBLES - A TIMETABLE MANAGEMENT SYSTEM"** is carried out by **Ms. ANUSHA PHANIRAJ** bearing USN **1KS20AI004** bonafide student of **K.S. Institute of Technology** in the partial fulfilment for the award of the **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University, Belagavi**, during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini-project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said degree for the Fifth semester.

Dr. Vaneeta M
Associate Prof & HOD, AIML Department

Dr. Dilip Kumar K
Principal/Director, KSIT

Ms. Lakshmi K K
Asst. Prof, Dept. Of AIML

Ms. Anu Mathews
Asst. Prof, Dept. Of AIML

Name of the Examiners Signature with date

1.
2.

ACKNOWLEDGEMENT

I take this opportunity to thank everyone involved in the successful implementation of this mini project. I would like to thank the college for providing me an opportunity to work on the mini project.

I take this opportunity to express my sincere gratitude to my college **K.S. Institute of Technology,** Bengaluru for providing the environment to work on this mini project.

I would like to express my gratitude to our **MANAGEMENT,** K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the support provided for carrying out this mini project work in college.

I would like to express my gratitude to **Dr. K.V.A Balaji, CEO**, K.S. Institute of Technology, Bengaluru, for his valuable guidance.

I would like to express my gratitude to **Dr. Dilip Kumar K**, **Principal/Director**, K.S. Institute of Technology, Bengaluru, for his continuous support.

I like to extend my gratitude to **Dr. Vaneeta M**, **Associate Professor and Head**, Department of Artificial Intelligence & Machine Learning, for providing very good facilities and all the support in successfully carrying out this Mini Project.

I also like to thank my Mini Project Coordinators, **Ms. Anu Mathews, Asst. Professor, Ms. Lakshmi K K, Asst. Professor,** Department of Artificial Intelligence & Machine Learning, for their help and support in successfully carrying out the Mini Project work.

I am also thankful to the teaching and non-teaching staff of the Artificial Intelligence & Machine Learning Department, KSIT for the help provided in completing this Mini Project.

ANUSHA PHANIRAJ

1KS20AI004

ABSTRACT

TIBBLES is a Timetable Generator DBMS that enables educational institutions to efficiently generate and manage their class schedules. It allows the administrators to create and organize class schedules, and assign teachers to courses. The system is user-friendly and makes it easy for the administrators to make changes or adjustments to the timetable as needed. It also allows the users to view their schedules and enter subject and time preferences. This system can generate timetables in an optimized manner to manage clashes and resolve them. The system is based on a database management system, which allows for the storage and retrieval of data related to the class schedule, including class times, subjects and teachers. Additionally, the system allows for the generation of reports for various purposes.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Tibbles is a Database Management Systems project designed to generate timetables for different semesters of various departments in a college taking in account subject and time preferences of available teaching faculty. By automating the scheduling process, the system can greatly improve the efficiency of the institution, reducing the amount of time and resources that are required to create schedules manually. The system can reduce the number of scheduling conflicts that occur, such as classes being scheduled at the same time or teachers being unavailable. The application of the timetable generator DBMS project is designed to be as simple as possible to minimise errors when entering data. The system is user-friendly and does not require any special knowledge to use it. Tibbles can generate error-free, secure, reliable, and fast schedules by automating the scheduling process and reducing conflicts. This makes it easy for users to use and ensures that the schedules generated are accurate and efficient.

1.2 PROBLEM STATEMENT

The main aim of the timetable generator DBMS project is to automate the scheduling process for educational institutions and to provide a user-friendly and efficient system for generating schedules without conflicts that meet the institution's specific needs.

1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is a system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organised and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database’s logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL codes are not completely portable among different database systems without adjustments.

1.5 HTML

HTML (Hypertext Markup Language) is the standard language used for creating web pages. It is a markup language that uses a set of tags to structure and format the content of a web page. HTML tags are used to create the layout, organise the content, and add multimedia elements such as images and videos. The structure and layout of a web page is defined using the HTML tags, such as headings, paragraphs, lists, links, images, and more. HTML also allows you to add interactive elements to a web page such as forms, buttons, and links. HTML also allows you to create dynamic web pages with the use of JavaScript and CSS. CSS (Cascading Style Sheets) is used to control the visual presentation of a web page, and JavaScript is used to create interactive and dynamic effects on a web page. HTML has evolved with time and the current version of HTML is HTML5, which includes new tags and features to create more interactive and multimedia-rich web pages. HTML5 also allows for better semantic meaning, making it easier for search engines and assistive technologies to understand the content of a web page.

1.6 DJANGO

Django is a high-level Python web framework that enables the rapid development of secure and maintainable websites. It follows the Model-View-Controller (MVC) architectural pattern and emphasises the reusability and "pluggability" of components. It has an Object-relational mapper (ORM) that allows you to interact with the database using Python code. Django uses a templating engine to separate the presentation logic from the business logic. It also has a built-in admin interface that allows you to easily manage the data in your application. It has built-in support for user authentication and authorization, including user registration, login, logout, and password reset. The URL routing system maps URLs to views, it also has a powerful forms library and a

middleware system that allows you to add functionality to your application. Additionally, it has built-in support for internationalisation and localization and testing framework. It has a large and active community, which means that there are many third-party packages and libraries available that can be easily integrated into your application.

CHAPTER 2

SYSTEM ANALYSIS

System analysis will be performed to determine if it is feasible to design information based on policies and

plans of the organisation and on user requirements and to eliminate the weakness of the present system.

• The new system should be cost effective.

• To expand management, improve productivity and services.

• To enhance user/ system interface.

• To improve information quality and usability.

• To upgrade systems reliability, availability, flexibility and growth potential.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

Processor : Intel i5 or above, AMD Ryzen 5 or above

Memory : 4GB or above

3.2 SOFTWARE REQUIREMENTS

Web Browser : Firefox 109, Google Chrome 109

Database support : MySQL 8.0

MySQL Server 8.0

MySQL Shell 8.0

MySQL Workbench

Python 3.11

Django 4.1.5

Operating system : Windows 11 / Ubuntu 22.04

3.3 TECHNOLOGY

HTML is used for the front end design. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document. SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
Django is a high-level, open-source web framework for building web applications using the Python programming language. It follows the Model-View-Controller (MVC) architectural pattern and emphasises reusability and "pluggability" of components. Django has built-in support for Object-relational mapping (ORM), templating, user authentication, URL routing, forms, middleware, internationalisation, localization and testing. It's widely used for building robust and maintainable web applications and has a large and active community.

CHAPTER 4

SYSTEM DESIGN

4.1 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. Entities may be characterised not only by relationships but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models. An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database, a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key". There is a tradition for ER/data models to be built at two or three levels of abstraction. Fig. 4.1 is the ER diagram for Tibbles project.

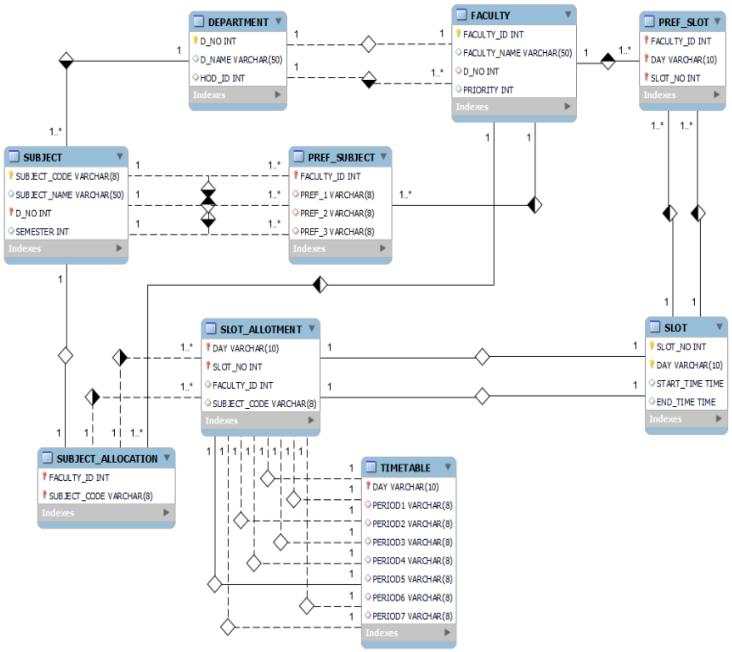

Fig. 4.1 Entity- Relationship Diagram

4.2 RELATIONAL SCHEMA DIAGRAM

The term “schema” refers to the organisation of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organise values in the database. The following diagram shows the schema diagram for the database. Fig 4.2. is the relational schema for Tibbles project.

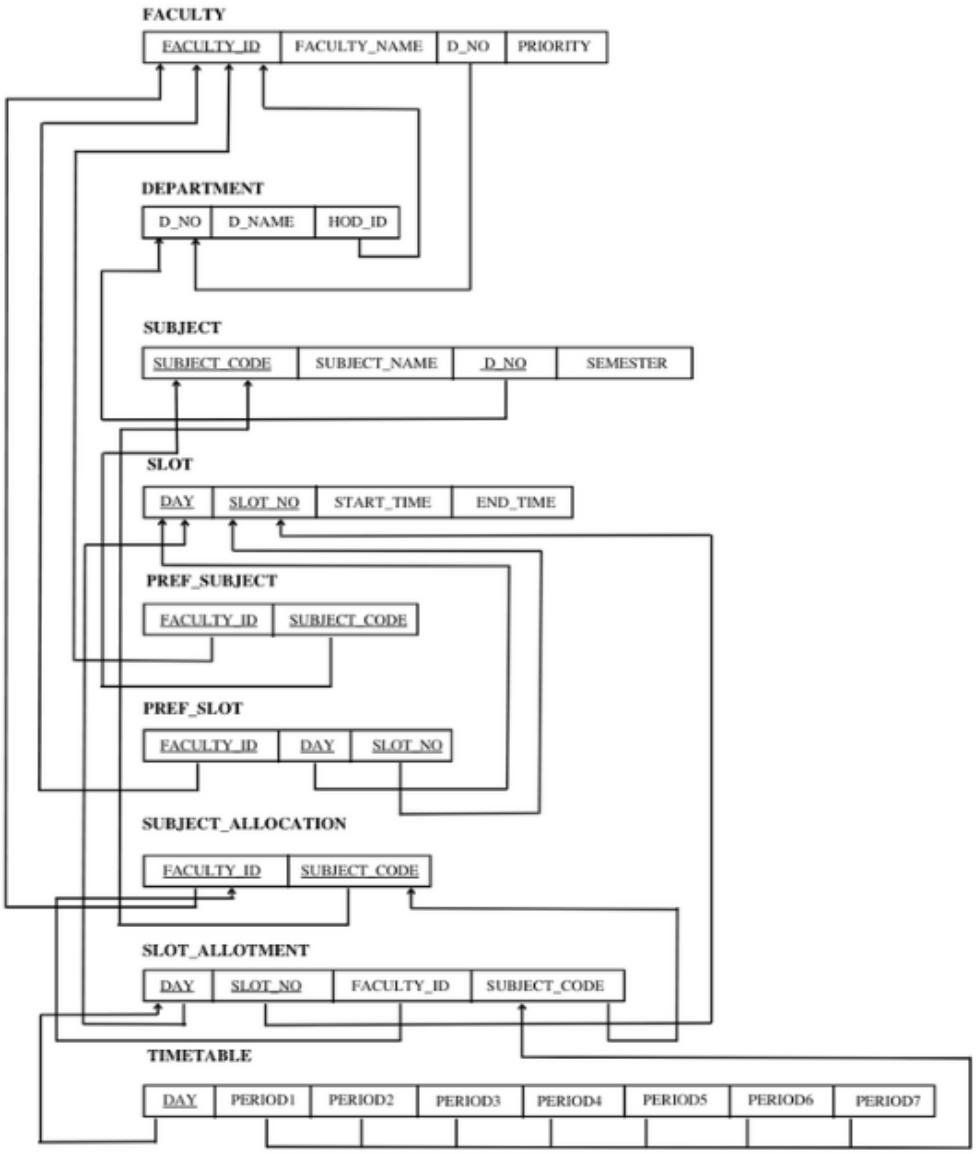

Fig 4.2 Relational Schema Diagram

CHAPTER 5

IMPLEMENTATION

5.1 DESCRIPTION OF TABLES

This database consists of tables:

FACULTY: This table stores faculty data.

- Faculty_ID: unique identifier for each faculty.
- Faculty_Name: name of the faculty.
- D_no: department ID of department to which the faculty belongs to
- Priority: each faculty is given a priority number (1 - Guest lecture, 2 - HOD, 3 - Assistant professor)

DEPARTMENT: This table stores department data.

- D_no: unique identifier for each department.
- D_Name: name of the department
- HOD_ID: ID of the faculty who is the head of the department.

SUBJECT: This table stores information of subjects offered by each department for each semester.

- Subject_code : unique code for each subject which indicates which department should handle it and in which semester that course belongs to.
- Subject_Name: name of the subject.
- Semester: each subject must be taught for a certain semester.
- Department: each subject belongs to one or more departments.

SLOT: This table stores data about day and timings.

- Slot_No: each slot number represents a certain time period of the day.
- Day: the day of the week.
- Start_time: start time of the slot.
- End_time: end time of the slot.

PREF_SLOT: This table stores slot preferences of faculty.

- Faculty_ID: identifier of faculty.

- Day: day of the week.
- Slot_no: preferred slot of the faculty.

PREF_SUBJECT: This table stores subject preferences of the faculty.

- Faculty_ID: identifier of faculty.
- Subject_code: code of the subject preferred by the faculty.

SUBJECT_ALLOCATION: This table holds data of subjects allotted to each faculty.

- Faculty_ID: identifier for faculty.
- Subject_code: code of the subject allocated to the faculty.

SLOT_ALLOTMENT: This table indicates slots allocated to subjects.

- Day: day of the week.
- Slot_no: time period of the day.
- Subject_code: code of the subject.
- Faculty_ID: identifier of the faculty handling the subject.

5.2 STORED PROCEDURES

PREFERENCE 1 ALLOTMENT:

```
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE fac_id INT;
  DECLARE subj_code VARCHAR(255);
  DECLARE prio INT;
  DECLARE cur CURSOR FOR SELECT faculty_id AS fac_id, priority AS prio FROM
faculty ORDER BY priority, faculty_id ASC;
  DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = TRUE;

  OPEN cur;
  read_loop: LOOP
        FETCH cur INTO fac_id, prio;
        IF done THEN
        LEAVE read_loop;
```

```
        END IF;
        SELECT pref_1 INTO subj_code FROM pref_sub WHERE faculty_id = fac_id;

        SELECT d_no INTO @dept FROM subject WHERE subject_code = subj_code LIMIT
1;
        IF NOT EXISTS (SELECT 1 FROM subject_allocation WHERE subject_code =
subj_code) THEN
                IF EXISTS (SELECT 1 FROM subject S, faculty F WHERE
S.SUBJECT_CODE = subj_code AND S.D_NO = F.D_NO AND S.D_NO = @dept AND
F.D_NO = @dept AND F.FACULTY_ID = fac_id) THEN
                IF @dept = 1 THEN
                INSERT INTO subject_allocation (faculty_id, subject_code) VALUES (fac_id,
subj_code);
                END IF;
        END IF;
        END IF;
  END LOOP read_loop;
  CLOSE cur;
END
```

PREFERENCE 2 ALLOTMENT:

```
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE fac_id INT;
  DECLARE subj_code VARCHAR(255);
  DECLARE prio INT;
  DECLARE cur CURSOR FOR SELECT faculty_id AS fac_id, priority AS prio FROM
faculty ORDER BY priority, faculty_id ASC;
  DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = TRUE;

  OPEN cur;

  read_loop: LOOP
```

```
        FETCH cur INTO fac_id, prio;
        IF done THEN
        LEAVE read_loop;
        END IF;
        SELECT pref_2 INTO subj_code FROM pref_sub WHERE faculty_id = fac_id;
        SELECT d_no INTO @dept FROM subject WHERE subject_code = subj_code LIMIT
1;
        IF NOT EXISTS (SELECT 1 FROM subject_allocation WHERE subject_code =
subj_code) THEN
                IF EXISTS (SELECT 1 FROM subject S, faculty F, department D WHERE
S.SUBJECT_CODE = subj_code AND S.D_NO = F.D_NO AND S.D_NO = @dept AND
F.D_NO = @dept AND F.FACULTY_ID = fac_id) THEN
        IF @dept = 1 THEN
                INSERT INTO subject_allocation (faculty_id, subject_code) VALUES (fac_id,
subj_code);
          END IF;
        END IF;
        END IF;
  END LOOP read_loop;
  CLOSE cur;
END
```

PREFERENCE 3 ALLOTMENT:

```
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE fac_id INT;
  DECLARE subj_code VARCHAR(255);
  DECLARE prio INT;
  DECLARE cur CURSOR FOR SELECT faculty_id AS fac_id, priority AS prio FROM
faculty ORDER BY priority, faculty_id ASC;
  DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = TRUE;
```

```
  OPEN cur;

  read_loop: LOOP
        FETCH cur INTO fac_id, prio;
        IF done THEN
        LEAVE read_loop;
        END IF;
        SELECT pref_3 INTO subj_code FROM pref_sub WHERE faculty_id = fac_id;
        SELECT d_no INTO @dept FROM subject WHERE subject_code = subj_code LIMIT
1;
        IF NOT EXISTS (SELECT 1 FROM subject_allocation WHERE subject_code =
subj_code) THEN
                IF NOT EXISTS (SELECT 1 FROM subject_allocation WHERE faculty_id =
fac_id GROUP BY Faculty_ID HAVING COUNT(Faculty_ID) >= 2) THEN
                IF EXISTS (SELECT 1 FROM subject S, faculty F, department D WHERE
S.SUBJECT_CODE = subj_code AND S.D_NO = F.D_NO AND S.D_NO = @dept AND
F.D_NO = @dept AND F.FACULTY_ID = fac_id) THEN
                IF @dept = 1 THEN
                INSERT INTO subject_allocation (faculty_id, subject_code) VALUES (fac_id,
subj_code);
                END IF;
                END IF;
        END IF;
        END IF;
  END LOOP read_loop;
  CLOSE cur;
END
```

REMAINING SUBJECTS ALLOTMENT:

```
BEGIN
        DECLARE done INT DEFAULT FALSE;
        DECLARE sub VARCHAR(255);
```

```
    DECLARE fac INT;
    DECLARE subject_cursor CURSOR FOR SELECT subject_code FROM subject
WHERE d_no = 1 AND subject_code NOT IN (SELECT subject_code FROM
subject_allocation);
    DECLARE faculty_cursor CURSOR FOR SELECT faculty_id FROM faculty WHERE
d_no = 1;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN subject_cursor;
    subject_loop: LOOP
    FETCH subject_cursor INTO sub;
    IF done THEN
       LEAVE subject_loop;
    END IF;
    SET done = FALSE;
    OPEN faculty_cursor;

    faculty_loop: LOOP
    FETCH faculty_cursor INTO fac;
    IF done THEN
        LEAVE faculty_loop;
    END IF;
    IF NOT EXISTS (SELECT 1 FROM subject_allocation WHERE faculty_id = fac
GROUP BY Faculty_ID HAVING COUNT(Faculty_ID) >= 2) THEN
        INSERT INTO subject_allocation (faculty_id, subject_code) VALUES (fac,
sub);
        SET done = TRUE;
    END IF;
    END LOOP;
    CLOSE faculty_cursor;
    END LOOP;
    CLOSE subject_cursor;
END
```

TIMETABLE FOR 3rd SEM AIML ALLOTMENT:

```
BEGIN
        DECLARE done INT DEFAULT FALSE;
        DECLARE sub VARCHAR(255);
        DECLARE d ENUM('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday');
        DECLARE sn INT;
        DECLARE cur CURSOR FOR SELECT subject_code, day, slot_no FROM
slot_allocation_sem3;
        DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = TRUE;

        OPEN cur;
        allocation_loop: LOOP
        FETCH cur INTO sub, d, sn;
          IF done THEN
        LEAVE allocation_loop;
        END IF;

        CASE sn
        WHEN 1 THEN
                UPDATE timetable3 SET period1 = sub WHERE d = day;
        WHEN 2 THEN
                UPDATE timetable3 SET period2 = sub WHERE d = day;
        WHEN 3 THEN
                UPDATE timetable3 SET period3 = sub WHERE d = day;
        WHEN 4 THEN
                UPDATE timetable3 SET period4 = sub WHERE d = day;
        WHEN 5 THEN
                UPDATE timetable3 SET period5 = sub WHERE d = day;
        WHEN 6 THEN
                UPDATE timetable3 SET period6 = sub WHERE d = day;
        WHEN 7 THEN
                UPDATE timetable3 SET period7 = sub WHERE d = day;
```

```
        WHEN 8 THEN
            UPDATE timetable3 SET period1 = sub, period2 = sub WHERE d = day;
        WHEN 9 THEN
            UPDATE timetable3 SET period3 = sub, period4 = sub WHERE d = day;
        WHEN 10 THEN
            UPDATE timetable3 SET period5 = sub, period6 = sub WHERE d = day;
        WHEN 11 THEN
            UPDATE timetable3 SET period6 = sub, period7 = sub WHERE d = day;
        WHEN 12 THEN
            UPDATE timetable3 SET period5 = sub, period6 = sub, period7 = sub WHERE
d = day;

        END CASE;
        END LOOP;
        CLOSE cur;
END
```

TIMETABLE FOR 5th SEM AIML ALLOTMENT:

```
BEGIN
         DECLARE done INT DEFAULT FALSE;
        DECLARE sub VARCHAR(255);
        DECLARE d ENUM('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday');
        DECLARE sn INT;
        DECLARE cur CURSOR FOR SELECT subject_code, day, slot_no FROM
slot_allocation_sem5;
        DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = TRUE;

        OPEN cur;
        allocation_loop: LOOP
        FETCH cur INTO sub, d, sn;
        IF done THEN
        LEAVE allocation_loop;
        END IF;
```

```
        CASE sn
        WHEN 1 THEN
                UPDATE timetable5 SET period1 = sub WHERE d = day;
        WHEN 2 THEN
                UPDATE timetable5 SET period2 = sub WHERE d = day;
        WHEN 3 THEN
                UPDATE timetable5 SET period3 = sub WHERE d = day;
        WHEN 4 THEN
                UPDATE timetable5 SET period4 = sub WHERE d = day;
        WHEN 5 THEN
                UPDATE timetable5 SET period5 = sub WHERE d = day;
        WHEN 6 THEN
                UPDATE timetable5 SET period6 = sub WHERE d = day;
        WHEN 7 THEN
                UPDATE timetable5 SET period7 = sub WHERE d = day;
        WHEN 8 THEN
                UPDATE timetable5 SET period1 = sub, period2 = sub WHERE d = day;
        WHEN 9 THEN
                UPDATE timetable5 SET period3 = sub, period4 = sub WHERE d = day;
        WHEN 10 THEN
                UPDATE timetable5 SET period5 = sub, period6 = sub WHERE d = day;
        WHEN 11 THEN
                UPDATE timetable5 SET period6 = sub, period7 = sub WHERE d = day;
        WHEN 12 THEN
                UPDATE timetable5 SET period5 = sub, period6 = sub, period7 = sub WHERE
d = day;
        END CASE;
        END LOOP;
        CLOSE cur;
END
```

CHAPTER 6

SNAPSHOTS

6.1 MAIN PAGE

Fig 6.1

The main page describes the project in 3 words and contains links to the login page and the about project page.

6.2 ABOUT PAGE

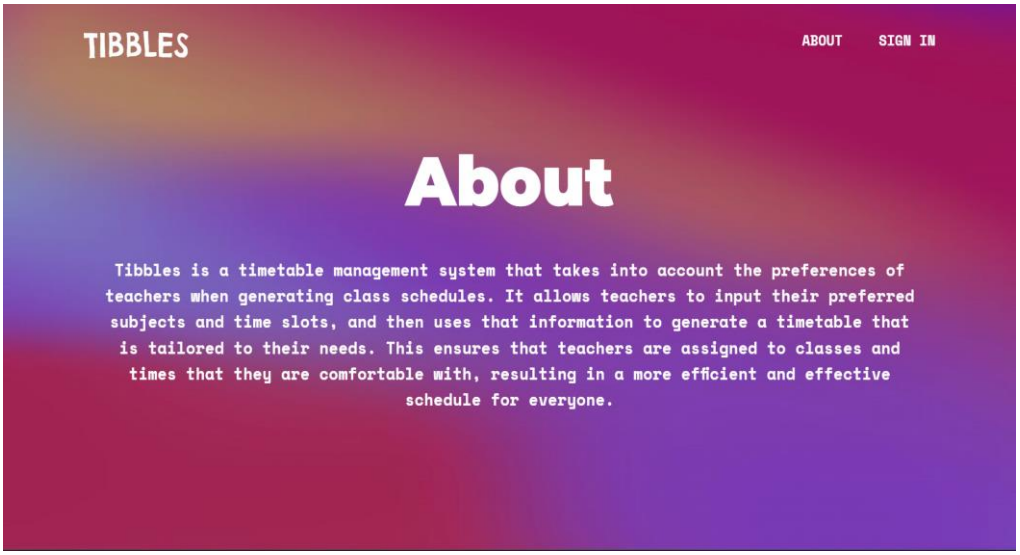

Fig 6.2

This page describes what Tibbles is about in a short and clear manner. It is a timetable management system.

6.3 LOGIN PAGE

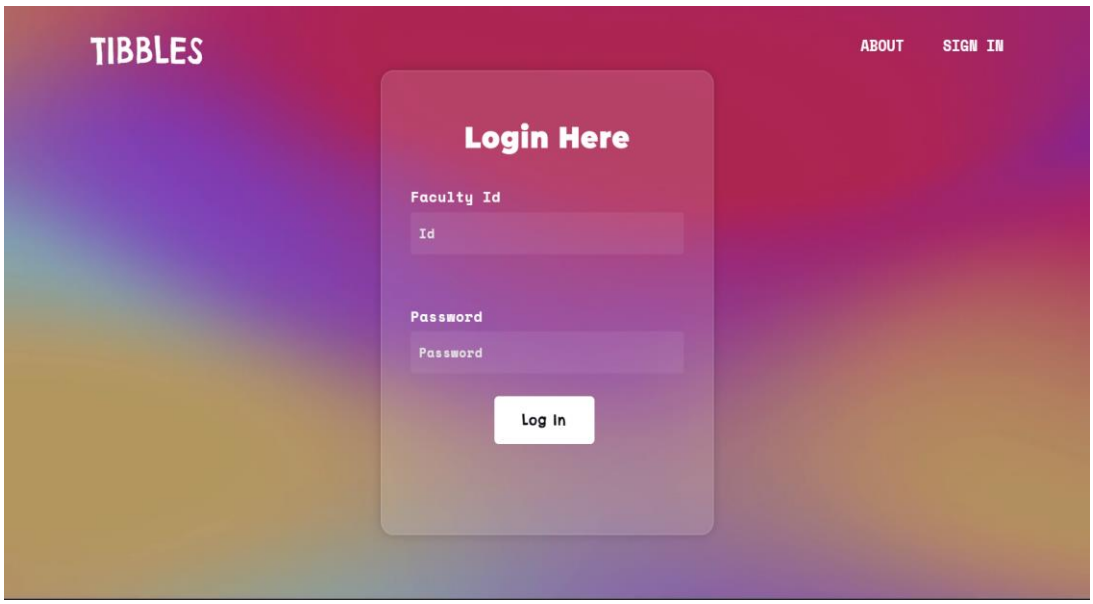

Fig 6.3

This is the login page for faculty. They need to log in using a unique faculty ID and password given by the admin.

6.4 SIGN UP PAGE

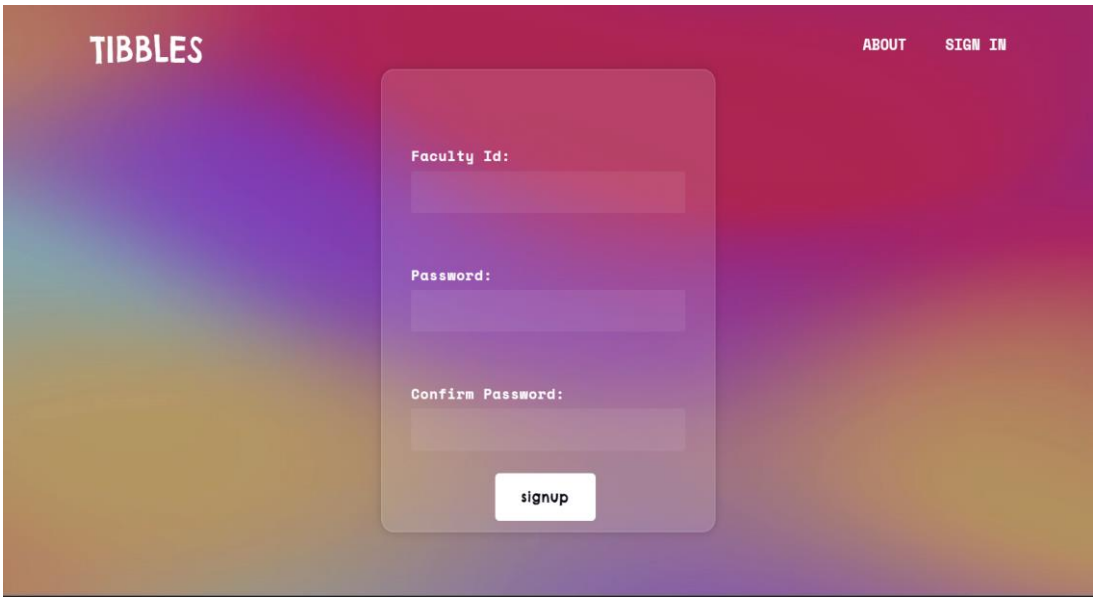

Fig 6.4

The signup page can be accessed by the admin(HOD) to add new faculty to the system. It contains faculty ID, password and confirm password fields.

6.5 ADMIN (HOD) HOME PAGE

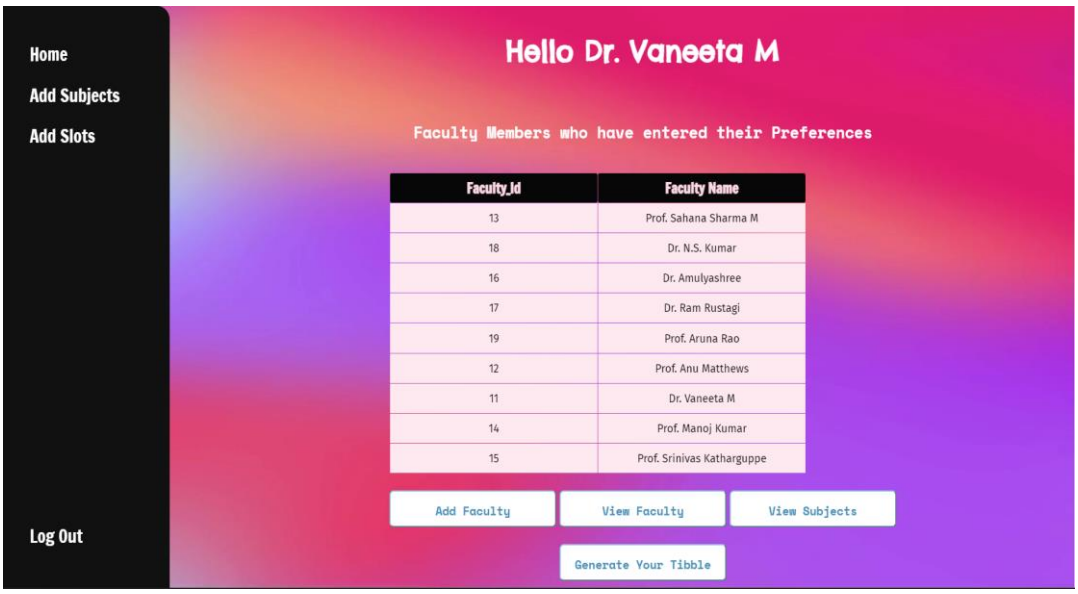

Fig 6.5

Admin (HOD) home page. HOD can view who has entered their preferences, the list of faculties in their department and subjects handled by their department.

6.6 FACULTY HOME PAGE

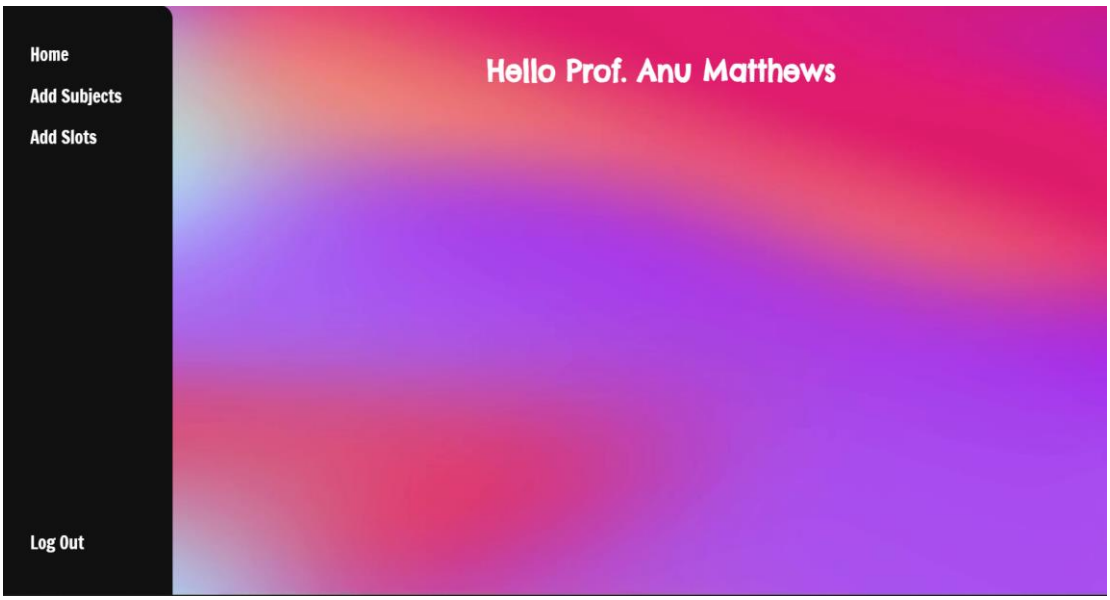

Fig 6.6

This is the faculty home page. It contains links to pages where faculty can add their subject and slot preferences.

6.7 ADD SUBJECT PREFERENCES PAGE

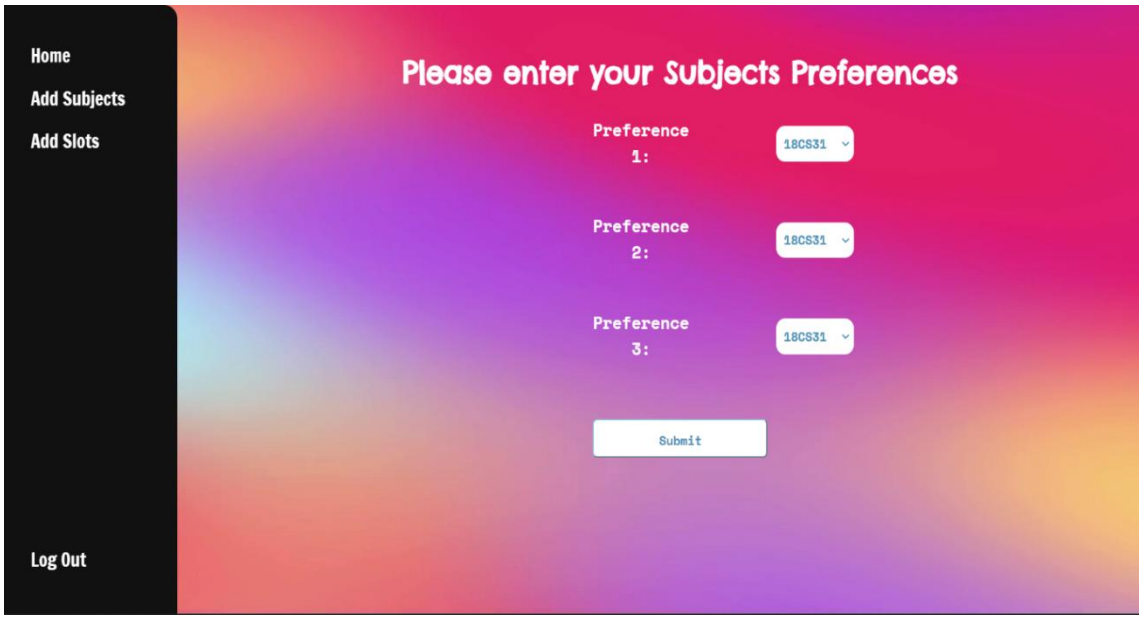

Fig 6.7

The subject preference page allows faculty to add 3 subjects of preference to the database. The timetable is scheduled based on the faculty's preference.

6.8 ADD SLOT PREFERENCES PAGE

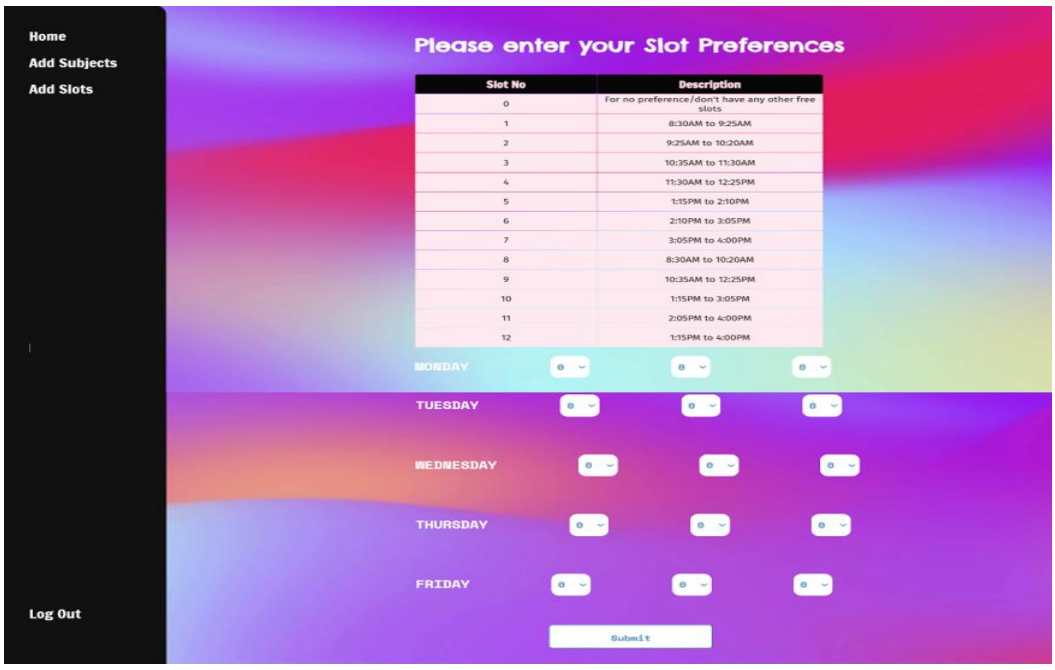

Fig 6.8

The slot preference page allows faculty to add 3-time slots per day of preference to the database. The timetable is scheduled based on the faculty's preference.

6.9 GENERATED TIMETABLE PAGE

DAY	1	2	3	4	5	6	7
MONDAY	18CS32	18CS33	18CS32	18CS33	18CSL37	18CSL37	18CSL37
TUESDAY	18CS32	18CS35	18CS32	18CS36	18CSL38	18CSL38	18CSL38
WEDNESDAY	18CS32	18CS36	18CS34	18CS33	18CS34	18KAK39	18CS34
THURSDAY	18KVK39	18CS36	None	None	18CS31	18CS31	None
FRIDAY	None	None	18CS35	18CS35	18CS31	18CS31	None

SUBJECT	SUBJECT CODE	FACULTY
Mathematics	18CS31	Prof. Srinivas Kathargуppe
Data Structures and Applications	18CS32	Dr. Vaneeta M
Analog and Digital Electronics	18CS33	Dr. Amulyashree
Computer Organization	18CS34	Dr. Vaneeta M
Software Engineering	18CS35	Dr. Ram Rustagi
Discrete Mathematical Structures	18CS36	Dr. Ram Rustagi
ADE Laboratory	18CSL37	Prof. Anu Matthews
DSA Laboratory	18CSL38	Dr. N.S. Kumar
Aadalitha Kannada	18KAK39	Prof. Krupesha
Vyavaharika Kannada	18KVK39	Prof. Krupesha

5th semester

Day	1	2	3	4	5	6	7
MONDAY	18CS53	18CIV59	18AI56	18AI56	18CSL58	18CSL58	18CSL58
TUESDAY	18AI55	18AI55	18CS53	18CS53	18AIL57	18AIL57	18AIL57
WEDNESDAY	18AI56	18CS53	18CS51	18CS53	18AI52	18AI52	18AI55
THURSDAY	18CS54	18CS51	None	18AI52	18CS51	18CS54	18CS54
FRIDAY	18AI52	None	None	None	18CS51	18AI52	None

SUBJECT	SUBJECT CODE	FACULTY
Python Programming	18AI52	Dr. N.S. Kumar
Principles of Artificial Intelligence	18AI55	Prof. Sahana Sharma M
Mathematics for Machine Learning	18AI56	Prof. Manoj Kumar
AI Laboratory	18AIL57	Prof. Anu Matthews
Enivronmental Studies	18CIV59	Prof. Srinivas Kathargуppe
Management and Entrepreneurship for IT	18CS51	Prof. Sahana Sharma M
Database Management Systems	18CS53	Prof. Manoj Kumar
Automata Theory and Computability	18CS54	Dr. Amulyashree
DBMS Laboratory	18CSL58	Prof. Aruna Rao

Fig 6.9

The final timetable for the 3rd and 5th semesters of the AIML department is displayed on this page along with a list of subjects and names of the faculty who handle the respective subjects.

CONCLUSION

This platform simplifies an administrator's job of allocating subjects and drawing a schedule for various semesters across departments under the constraints of faculty preference. Once all the faculty from a certain department have entered their preferences of subject and timings, the admin has to click the "Generate timetable" button and the system will generate a timetable that is feasible for faculty based on their preferences while ensuring that clashes don't appear in schedules across different semesters. This project was developed using HTML5, CSS, MySQL and Django. The goals achieved by this project are:

- Centralized database
- User-friendly environment.
- Efficient manner of storing and managing all faculty preferences in one place.
- Only authorized personnel can generate timetable
- Most preferences are taken into consideration under constraints to construct schedule.

FUTURE SCOPE AND ENHANCEMENT

The future scope and enhancement of this project could include several possibilities such as:

1. Integration with other systems such as student information systems, room booking systems, and calendar systems to improve the efficiency and accuracy of the timetable generation process.
2. Incorporating machine learning algorithms to optimize the timetable based on factors such as student enrolment, class size, and historical data.
3. Allowing multiple stakeholders, such as department heads, to collaborate and approve the final schedule. Incorporating real-time updates and notifications to keep users informed of any schedule changes or conflicts.

REFERENCES

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition,2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill
3. https://developer.mozilla.org/en-US/
4. https://www.djangoproject.com/
5. https://dev.mysql.com/doc/