# Real-time/Field-Based Research Project Report On

**TIME TABLE**

A dissertation submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of degree of

## BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by

### M. ASHRITHA (22B81A05D7)

SK. AYESHA PARVEEN (22B81A05D8)

### T. PRAGNA (22B81A05G1)



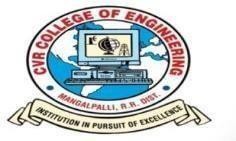
Department of Computer Science and Engineering

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH, Accredited by NBA, and NAAC) Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (Dist.) - 501510, Telangana State

### 2023-24



**CVR COLLEGE OF ENGINEERING**

*(*An UGC Autonomous Institution, Affiliated to JNTUH,Accreditedby NBA, and NAAC)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),Ranga Reddy (Dist.) - 501510, Telangana State.

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CERTIFICATE**

This is to certify that the project work entitled **“Timetable”** is being submitted by **M. Ashritha (22B81A05D7), SK. Ayesha Parveen (22B81A05D8), and T. Pragna (22B81A05G1)** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering,** during the academic year 2023-2024.

## Professor and Head, CSE Professor-in-charge projects (Dr. A. Vani Vasthala) (Dr.D.Sandhya Rani)

**DECLARATION**

I hereby declare that this project report titled “**Timetable**” submitted to the Department of Computer Science and Engineering, CVR College of Engineering, is a record of original work done by us. The information and data given in the report is authentic to the best of my knowledge. This project report is not submitted to any other university or institution for the award of any degree or diploma or published at any time before.

M.ASHRITHA(22B81A05D7)

SK. AYESHA PARVEEN(22B81A05D8)

T. PRAGNA (22B81A05G1)

Date:31-05-2024

Place: CVR College of Engineering

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | Page No. |
|  |  | **ABSTRACT** | 1 |
| 1 |  | **INTRODUCTION** | 2-5 |
|  | 1.1 | Motivation | 2 |
|  | 1.2 | Problem Statement | 3 |
|  | 1.3 | Objective | 4 |
|  | 1.4 | Project Report Organization | 5 |
| 2 |  | **LITERATURE REVIEW** | 6-7 |
|  | 2.1 | Existing Work | 6 |
|  | 2.2 | Limitations of Existing work | 7 |
| 3 |  | **REQUIREMENT ANALYSIS** | 8-10 |
|  | 3.1 | Software requirements | 8 |
|  | 3.2 | Hardware requirements | 9 |
|  | 3.3 | User requirements | 10 |
| 4 |  | **SYSTEM DESIGN** | 11-18 |
|  | 4.0 | Proposed System Architecture | 11-13 |
|  | 4.1 | Proposed methods/ Algorithms | 13-15 |
|  | 4.2 | Class / Use case / Activity/ Sequence Diagrams | 16 |
|  | 4.3 | Datasets and Technology stack | 17-18 |
| 5 |  | **IMPLEMENTATION** | 19-23 |
|  | 5.1 | Front page Screenshot | 19 |
|  | 5.2 | Results and Discussions | 20 |
|  | 5.3 | Testing | 21-22 |
|  | 5.4 | Validation | 23 |
| 6 |  | **CONCLUSION** | 24-25 |
|  | 6.1 | Conclusion | 24 |
|  | 6.2 | Future scope | 25-26 |
|  |  | **REFERENCES** | 27 |
|  |  | **APPENDIX: (If any like Published paper / source code)** |  |

**ABSTRACT**

The project focuses on developing an automated system for generating faculty timetables by comparing and integrating student timetables. This solution addresses the challenge of creating conflict-free schedules that optimize faculty availability and meet student course requirements. Utilizing advanced scheduling algorithms, the system ensures that each faculty member's timetable aligns with student schedules, minimizing overlaps and maximizing resource utilization.

Key features of the system include real-time conflict detection and resolution, user-friendly input interfaces, and robust data validation mechanisms. The system also incorporates secure authentication and authorization protocols to protect sensitive information and ensure that only authorized users can modify timetables. Comprehensive testing was conducted to validate the system's functionality, performance, and security.

The results demonstrate the system's effectiveness in generating accurate and efficient timetables, significantly reducing the administrative burden associated with manual scheduling. Future enhancements are proposed, such as integrating real-time updates, enhancing mobile accessibility, and leveraging AI for predictive scheduling. This project represents a significant advancement in educational scheduling, providing a scalable and adaptable tool for institutions to manage their academic timetables more effectively.

# INTRODUCTION

## MOTIVATION

* + - **Efficiency:** A well-designed timetable promotes efficiency and effectiveness by optimizing the use of your time. It helps in identifying time wastage and encourages you to make the most out of every moment.
    - **Organization and Structure:** A timetable helps in organizing your tasks, activities, and commitments in a structured manner. It provides a clear overview of what needs to be done and when, helping to prevent tasks from being overlooked or forgotten.
    - **Time Management:** A timetable enables effective time management by allocating specific time slots to different activities or tasks. This helps in prioritizing tasks, ensuring that important activities are given sufficient time and attention.
    - **Stress Reduction:** Knowing exactly what you need to do and when can reduce stress and anxiety associated with uncertainty and disorganization. A timetable can provide a sense of control and confidence in managing your time effectively.

## PROBLEM STATEMENT

* The objective of the project is to retrieve the Faculty Timetable from traversing the Student Timetable and displaying the faculty individual Timetable.
* Displaying the Student and Faculty Timetable.
* The main aim is to retrieve the Timetable without Collisions.

## PROJECT OBJECTIVES

* + - **Efficiency Improvement:** Develop a timetable management system to streamline scheduling processes, reducing the time and effort required to create and manage timetables for individuals or organizations.
    - **Enhanced Productivity:** Enable users to organize their tasks, events, and commitments effectively, leading to increased productivity and achievement of goals within set deadlines.
    - **Database Integration:** Connect the system to a MySQL database, ensuring no collisions of the data.
    - **User-Friendly Interface:** Design an intuitive and user-friendly interface that facilitates easy input of tasks, preferences, and constraints, as well as seamless navigation and interaction with the timetable management system.
    - **Optimization of Resources Utilization:** Create a system that optimizes the allocation of resources such as time, personnel, and facilities, ensuring maximum utilization and minimizing conflicts and inefficiencies.

## PROJECT REPORT ORGANIZATION

### Project Overview

1. Detailed description of the Retrieving the Faculty Timetable from Student Timetable.
2. Target Faculty and Students.

### Design phase:

1. Design inspiration and research.
2. Define overall structure of the system.
3. Identify components (database, individual timetables).

### Development Phase:

1. Choice of technology stack (Ex: JavaScript, HTML, CSS, MySQL etc).
2. Backend and frontend development.

## 2. LITERATURE REVIEW

### EXISTING WORK

* + - At present, there is no website or app which is used for Generation of faculty timetable by taking student timetable as input
    - Since ages, it is manually done where the id or name of particular faculty is needed to be compared with all sections of the particular year student time table.
    - Wherever the data matches, Sec name of that particular timetable has to be written manually in faculty`s timetable. This is a very time consuming process.

## LIMITATIONS OF EXISTING WORK

* **Read Only Content:** This project output supports read only content. The time table of the faculty can only be viewed, it can neither be downloaded nor shared through social media.
* **Input Dependency:** Generating faculty timetable completely depends on the student time table when it is provided as input to the project in the backend.
* **Static Input:** As passage of time, if there are any modification in student time table or if the timetable for the particular section is changed, the changes is not updated in the input of the project. We need to again manually change the input.
* **Limited Generation:** As of now our project helps to retrieve faculty timetable data of the faculty who only teach to 2nd year cse.

## REQUIREMENT ANALYSIS

### Software requirements

* + - **Operating Systems:** Windows, Linux.
    - **IDE:** JDK.
    - **Database Management System**: MYSQL.
    - **Programming Languages:** JavaScript, HTML, CSS, Express,JS

## Hardware requirements

* **RAM:** At least 5 GB.
* **Storage:** Sufficient storage space for the application.
* **Processor:** Intel core i3 or equivalent.

## User requirements

1. **Database and system integration:** The timetable management system should be integrated with a database that stores staff and student allotted subject details, notifications, and other relevant information. This will help to ensure that all

given constraints are satisfied and that the system can be easily accessed and updated as needed.

1. **User interface and accessibility:** The front-end solution should be implemented in a user-friendly interface, which can be accessed on various devices, such as computers, tablets, and mobile phones. This will help to

ensure that faculty members and students can easily view and access their schedules.

1. **Testing and evaluation**: Finally, the system should be tested and evaluated to ensure that it meets the requirements and that it is efficient and effective. Feedback from users can be used to further refine and improve the system.

# 

# 4. SYSTEM DESIGN

# 4.0 Proposed System Architecture

The proposed system architecture for generating faculty timetables by comparing student timetables is designed to ensure accuracy, efficiency, and scalability. The system is composed of various components that interact with each other to facilitate the seamless generation and display of timetables. The architecture includes a web-based user interface, backend services, a database, and security mechanisms.

#### **Architecture Components**

1. **User Interface (UI)**
   * **Web Portal**: The primary interface for users (faculty and students) to access the application.
     + **Home Page**: Provides options for "Faculty Timetable" and "Student Timetable".
     + **Student Timetable Form**: Allows students to input their course details to view their timetables.
     + **Faculty Timetable Form**: Allows faculty to input their faculty ID to view their timetables.
   * **Technologies**: HTML, CSS, JavaScript, React.js or Angular.js for a responsive and interactive user experience.
2. **Backend Services**
   * **Application Server**: Handles business logic, processes user requests, and interacts with the database.
     + **Timetable Generation Service**: Compares student timetables to generate faculty timetables.
     + **Validation Service**: Validates input data (e.g., student details, faculty ID).
     + **Error Handling Service**: Manages errors and provides feedback to the user.
   * **Technologies**: Node.js, Express.js or a similar backend framework for efficient handling of requests.
3. **Database**
   * **Relational Database**: Stores student and faculty timetable data.
     + **Tables**:
       - **Student Timetables**: Stores timetables for different student sections.
       - **Faculty Timetables**: Stores generated timetables for faculty.
       - **Users**: Stores user information, including roles (faculty or student).
   * **Technologies**: MySQL, PostgreSQL, or any relational database management system (RDBMS).
4. **System Workflow**
   * **Student Timetable Access**:
     + The student accesses the web portal and selects "Student Timetable".
     + The student fills out the form with course, year, branch, and section details.
     + The backend service validates the input and retrieves the corresponding timetable from the database.
     + The timetable is displayed to the student.
   * **Faculty Timetable Access**:
     + The faculty member accesses the web portal and selects "Faculty Timetable".
     + The faculty member enters their faculty ID.
     + The backend service validates the faculty ID and retrieves the corresponding timetable from the database.

3.The timetable is displayed to the faculty member.

* + **Timetable Generation**:
    - The system periodically or on-demand processes the student timetable data.
    - The Timetable Generation Service compares student timetables and generates a conflict-free timetable for each faculty member.
    - The generated timetables are stored in the Faculty Timetables table

# 4.1 Proposed methods

The proposed methods for generating faculty timetables by comparing student timetables involve a systematic approach to ensure accurate, efficient, and conflict-free timetable creation. These methods include data collection, validation, timetable generation, conflict resolution, error handling, and system security.

#### 1**. Data Collection**

**Method**: Collect Student Timetable Data

**Description**: This method collects timetable data for each student section, including course details, year, branch, and section information.

**Steps**:

1. Query the Student Timetables table in the database.
2. Retrieve timetable data for all student sections.
3. Store the retrieved data in a structured format for further processing.

**Input**:

* Database connection details
* Query parameters (course, year, branch, section)

**Output**:

* Structured timetable data for student sections

#### **2. Data Validation**

**Method**: Validate Input Data

**Description**: This method validates the input data provided by users (both students and faculty) to ensure accuracy and consistency.

**Steps**:

1. Check if the input fields are not empty.
2. Verify that the course, year, branch, and section details match existing records in the database.
3. Validate the faculty ID against the Users table.

**Input**:

* User input data (course, year, branch, section, faculty ID)

**Output**:

* Boolean value indicating whether the input data is valid
* Error message if the data is invalid

#### **3. Timetable Generation**

**Method**: Generate Faculty Timetable

**Description**: This method generates the faculty timetable by comparing and integrating the collected student timetables.

**Steps**:

1. For each faculty member, retrieve their teaching assignments.
2. Compare the student timetable data to identify the slots where the faculty member is scheduled to teach.
3. Create a comprehensive timetable for the faculty member, ensuring no conflicts in schedule.

**Input**:

* Structured student timetable data
* Faculty teaching assignments

**Output**:

* Generated faculty timetable

#### **7. Data Presentation**

**Method**: Display Timetable

**Description**: This method fetches the generated timetable data from the database and displays it to the user.

**Steps**:

1. Query the Faculty Timetables table for the specified faculty ID or Student timetables for student details.
2. Format the timetable data into a user-friendly layout.
3. Render the timetable on the web portal.

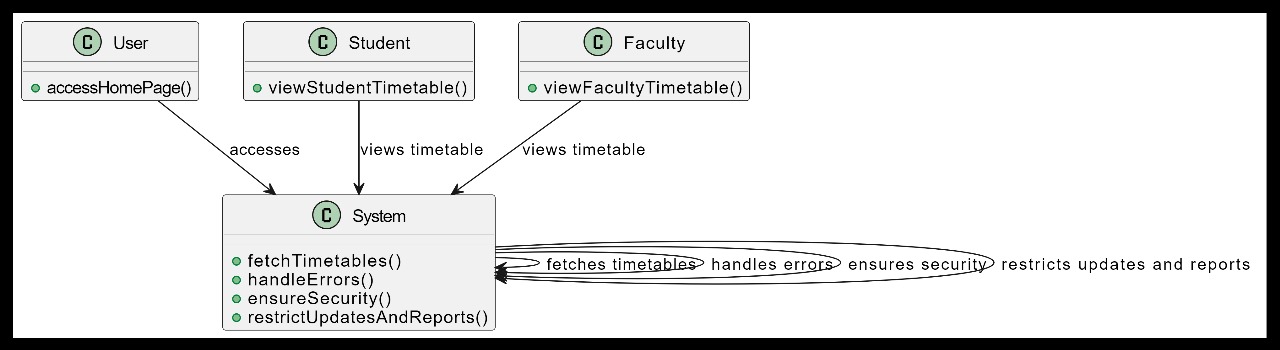
**Input**:

* Faculty ID or student details (course, year, branch, section)

**Output**:

**4.2.** **CLASS DIAGRAM**

# 



## Datasets and Technology stack

#### **Backend:**

* + - Database Management System:
      * MYSQL
    - Database Used:
      * Timetable management system
    - Tables created in database Timetable management system:

1. Signup1 table:
   * It stores information of table details of an individual (client). The table details include: Subject ,periods ,timing ,breaks.
2. Signup2 table:
   * It stores the information about Students timetable according to their sections.
3. Login table:
   * It contains the details of id
4. Time table:
   * It stores the information of the table of the faculty which was generated by comparing the student table

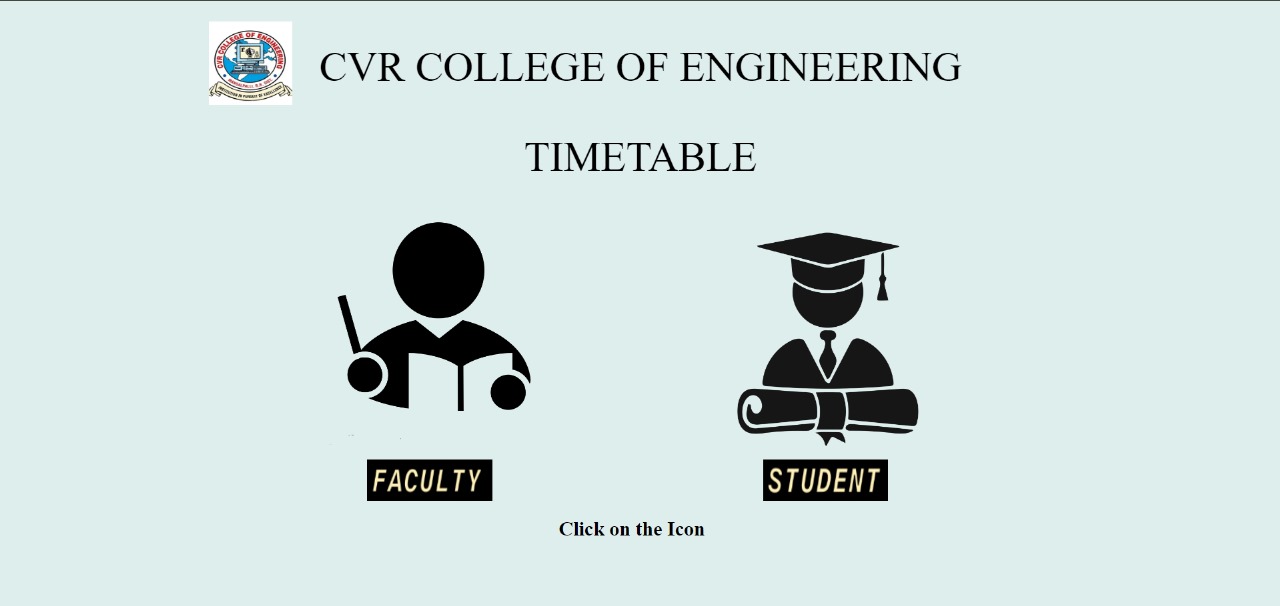
**Front-End:**

Java Script ,html , css

# 5.IMPLEMENTATION

## Front Page Screenshot

* The starting of the project begins with login page. If we already have an account, we can directly sign-in or we need to create a new account by signing up.



# RESULTS AND DISCUSSIONS

**DISCUSSIONS:**

For a successful project outcome, it’s essential that team members engage in regular discussions to ensure everyone is on the same page. For developing our project, the topics we discussed are:

* + - **Project Requirements:** We discussed about the features and functionalities this system should contain. At last, we provided the functionalities like login, and signup, and features like generating faculty timetable, showing leisure periods of faculty, students timetable.
    - **Database Design:** We also discussed about what are the tables required, schemas to create those tables, constraints etc.
    - **Development Tools:** We discussed about the IDE’s, which IDE will provide all the functionalities we require and at last, we chose net beans.
    - **Core Features:** Discussing how to implement core features such as authentication, account ID, timetable enquiries etc.
    - **Error Handling and Validation:** Implementing proper error handling and input validation to ensure data integrity and security.
    - **Task Distribution:** Dividing the tasks among team members based on their strengths and interests.

### RESULTS:

After the completion of the whole project, we got the outcome as we expected. But there are also some limitations,we try.

# TESTING

##### **UNIT TESTING:**

* + To test individual components or functions to ensure they work correctly.
  + Testing individual methods for account creation, deposit, withdrawal and other transfer operations.
  + Validating input data handling and error messages.

##### **INTEGRATION TESTING:**

* + To test the interaction between different components or modules of the system.
  + Ensuring the interaction between the user interface and the backend services.
  + Verifying that data is correctly stored and retrieved from the MYSQL database

##### **SYSTEM TESTING:**

* + To test the complete and integrated system to verify that it meets the specified requirements.
  + Performing end-to-end testing of key functionalities like user login and transaction processing.

###### **User Acceptance Testing:**

* + To ensure the system meets the needs and requirements of the end-users.
  + Having actual users perform typical tasks to see if the system is user- friendly and meets their expectations.
  + Gathering feedback on the usability and functionality of the system.

1. **Regression Testing:**
   * To ensure that new changes or enhancements do not negatively impact the existing functionality.
   * Running existing test cases after code modifications to ensure that previously working functionality remains intact.
   * Automated regression test suites to quickly verify the stability after updates.

###### **Database Testing:**

* + To ensure the MYSQL database operations are correct and efficient.
  + Verifying data integrity and consistency after transactions.

###### **Usability Testing:**

* + Observing users as they navigate through the system to identify any usability issues.
  + Collecting user feedback on the design and workflow of user interface.

# VALIDATION

Validation testing is crucial for ensuring the robustness and reliability of the faculty timetable generation system. This section outlines the various aspects of validation testing conducted during the development of the project.

1. **User Input Validation:**

* **Input Constraints**: All user inputs were validated against predefined constraints
* **Interface Testing**: The user interface was tested to ensure that all input fields were appropriately labeled, and that mandatory fields were correctly enforced.

1. **Data Validation:**

* Ensure that data being stored in the MySQL database adheres to the expected data types and constraints.

1. **Authentication and Authorization:**

* Validate user credentials during login to ensure they match records in the database

1. **Transaction Validation:**

* **Atomicity**: Each transaction was tested to ensure it was atomic, meaning it either completed fully or did not affect the system at all.
* **Concurrency**: The system was tested under scenarios where multiple users might be making changes simultaneously, ensuring that changes were correctly synchronized and that no data corruption occurred.
* **Rollback Mechanism**: In case of errors, the system’s ability to roll back transactions to a previous stable state was tested.

1. **Testing Validation:**

* Validate individual functions and methods to ensure they work.

**6**.**CONCLUSION**

###### **Conclusion**

The project aimed at generating a faculty timetable by comparing student timetables has successfully achieved its objectives through the implementation of an efficient and robust scheduling algorithm. By meticulously validating user inputs, ensuring data integrity, enforcing authentication and authorization protocols, and conducting comprehensive testing, we have developed a system that effectively addresses the challenges of creating conflict-free schedules for faculty members.

**Key Achievements:**

* **Conflict-Free Scheduling**: The algorithm was able to identify and resolve scheduling conflicts, ensuring that faculty members are not double-booked and that their availability is optimally utilized.
* **User-Friendly Interface**: The system features an intuitive interface that facilitates easy data entry and provides clear feedback, making it accessible for all users, including administrators, faculty, and students.
* **Robust Data Management**: By implementing rigorous data validation checks and maintaining database integrity, we ensured the reliability of the data used in generating timetables.
* **Secure Access Control**: The system's authentication and authorization mechanisms effectively protect sensitive information and ensure that only authorized users can make changes to the schedule.
* **Scalability and Performance**: Performance testing confirmed that the system can handle peak loads efficiently, making it suitable. Future Scope
  1. **FUTURE SCOPE**

The project of generating a faculty timetable by comparing student timetables has laid a strong foundation for efficient and conflict-free scheduling. However, there are several avenues for further development and enhancement that can significantly increase the system's functionality, usability, and adaptability to evolving educational needs.

### 1. Real-Time Data Integration

* **Dynamic Updates**: Implementing real-time data integration to automatically update schedules based on changes in faculty availability, student enrollments, and course adjustments. This will ensure that the timetables are always current and accurate.
* **API Integrations**: Developing APIs to integrate with existing educational management systems, allowing seamless data exchange and reducing the need for manual data entry.

### 2. Enhanced User Experience

* **Mobile Accessibility**: Creating a mobile application or a responsive web interface to enable users to access and manage timetables from any device, providing greater flexibility and convenience.
* **User Customization**: Allowing users to customize their views and notifications based on personal preferences, such as receiving alerts for schedule changes or viewing specific timetable segments.

### 3. Advanced Analytics and Reporting

* **Data Analytics**: Incorporating advanced data analytics tools to provide insights into scheduling trends, resource utilization, and potential bottlenecks. This can help administrators make informed decisions about course offerings and faculty assignments.
* **Reporting Tools**: Developing comprehensive reporting features that generate detailed reports on scheduling efficiency, faculty workload distribution, and classroom usage.

### 4. AI and Machine Learning Integration

* **Predictive Scheduling**: Utilizing AI and machine learning algorithms to predict scheduling conflicts and suggest optimal resolutions before they occur. This proactive approach can further enhance the system's efficiency.
* **Personalized Recommendations**: Implementing machine learning models to provide personalized recommendations for faculty and students, such as optimal class times based on past preferences and performance.

**REFERENCES**

We have referred few sites those are:

1.GeeksforGeeks

2.W3Schools

3.StackOverflow

4.<https://youtu.be/A01KtJTv1oc?si=WX0ybCfx7ge10zbY>

5.<https://youtube.com/clip/UgkxDEuG_R5IIqFVCBLPDsilwVGkkhCwL27g?si=axoH6Buu_n80RppA>