BMCC TRMS

BMCC: Brihan Maharashtra College of Commerce

TRMS: Teacher Resource Management System

Pratham Punjabi: TY-BBA(CA) 59

Hrutvik Dhotre: TY-BBA(CA) 20

Rohit Bodhankar: TY-BBA(CA) 11

TY-BBA(CA) Semester VI Project-BMCC, Pune

Introduction

BMCC TRMS (Teacher Resource Management System) is an innovative online platform revolutionizing administrative processes for educators. From attendance tracking to lecture scheduling, BMCC TRMS automates tasks, increasing efficiency and saving valuable time for teachers. Users can effortlessly record attendance, generate PDF reports, manage lecture timetables, and book facilities with ease.

Additionally, BMCC TRMS streamlines file management, group list generation, and task automation, enabling educators to focus more on teaching and less on administrative burdens. With its user-friendly interface and emphasis on efficiency, BMCC TRMS empowers educators to thrive in their roles and maximize productivity.

Key Features

1.**My Tasks (Add, Complete, Delete, Undo)**: Effortlessly manage your daily responsibilities. Add, complete, and delete tasks with ease, and even undo actions if needed, ensuring you stay organized and focused.

2.**Attendance (Mark, View)**: Simplify attendance tracking with intuitive marking and viewing options. Record attendance efficiently and access records conveniently whenever necessary.

3.**Lab (Book, View, Delete)**: Streamline lab facility bookings with just a few clicks. Easily view availability, make reservations, and manage bookings effectively. Delete or cancel bookings as needed to accommodate changing requirements.

4.**Hall (Book, View, Delete)**: Reserve halls for events or lectures seamlessly. Check availability, book suitable venues, and manage bookings effortlessly. Easily delete or cancel bookings as needed to accommodate changing requirements.

5.**Timetable (Create, View)**: Create personalized timetables tailored to your schedule. View your timetable for a quick overview of your day or week, ensuring optimal organization and productivity.

6.**Group List Generator**: Simplify group management with a versatile list generator. Quickly create group lists based on customizable criteria, saving time and effort in organizing group activities or assignments.

7.**Files (Upload, View, Download, Delete)**: Manage your files efficiently with versatile options. Upload, access, and download files as needed, and easily delete documents when required, ensuring seamless file management.

Objectives

1.**Enhance Efficiency**: Streamline administrative tasks such as attendance tracking, timetable management, and facility bookings to save time and resources for educators.

2.**Improve Organization**: Provide tools and features that help educators stay organized, manage tasks effectively, and access necessary resources conveniently.

3.**Increase Productivity**: Empower educators to focus more on teaching and student engagement by automating routine tasks and reducing administrative burdens.

4.**Facilitate Collaboration**: Foster collaboration and communication among educators by providing features for group management, file sharing, and scheduling.

5.**Improve Accessibility**: Ensure that essential information, resources, and tools are easily accessible to educators, regardless of their location or device.

6.**Enhance Data Management**: Facilitate the collection, storage, and analysis of data related to attendance, scheduling, and resource utilization to inform decision-making and improve educational outcomes.

7.**Enhance Student Experience**: Ultimately, aim to improve the overall educational experience for students by providing educators with the tools and support they need to excel in their roles.

Scope

BMCC TRMS can be utilized in various educational institutions including:

1.**Schools**: Both public and private schools, spanning from elementary to high school levels, can benefit from BMCC TRMS to streamline administrative tasks and enhance teaching efficiency.

2.**Universities and Colleges**: Higher education institutions, such as universities, colleges, and vocational schools, can utilize BMCC TRMS to manage complex scheduling, resource allocation, and communication needs across multiple departments and campuses.

3.**Online Learning Platforms**: Virtual schools, online academies, and distance learning programs can integrate BMCC TRMS to facilitate attendance tracking, file management, and collaboration among remote educators.

4.**Specialized Institutions**: Institutions offering specialized education, such as technical schools, language institutes, or art academies, can adapt BMCC TRMS to their unique administrative requirements and instructional methodologies.

5.**Educational Networks**: Educational networks comprising multiple institutions can implement BMCC TRMS to standardize administrative processes and promote efficiency across member institutions.

Scalability

1.**Platform Adaptability**: BMCC TRMS can be designed to be platform-agnostic, allowing it to run smoothly across various devices and operating systems. This includes not only desktop and laptop computers but also mobile devices such as smartphones and tablets. By ensuring compatibility with mobile platforms, educators can access the system anytime, anywhere, enhancing flexibility and convenience.

2.**Mobile App Development**: To further enhance accessibility and user experience, BMCC TRMS can be extended to include a dedicated mobile application. This app can provide educators with on-the-go access to essential features such as attendance tracking, task management, and file sharing. A mobile app can also enable push notifications for important updates and reminders, keeping educators informed and engaged even when they're away from their desks.

3.**Feature Expansion**: As educational needs evolve, BMCC TRMS can be scalable in terms of feature expansion. This includes the ability to add new functionalities and modules in response to user feedback and emerging requirements. For example, adding communication features such as messaging or discussion forums can facilitate collaboration among educators and students, promoting interaction and knowledge sharing within the platform.

4.**Scalable Infrastructure**: The underlying infrastructure supporting BMCC TRMS can be designed to accommodate growth in the user base and data volume. This may involve scalable cloud-based hosting solutions that can dynamically adjust resources based on demand, ensuring optimal performance and reliability even during peak usage periods.

5.**Integration Capabilities**: BMCC TRMS can facilitate seamless integration with external systems and services. This includes integration with learning management systems (LMS), student information systems (SIS), and other educational platforms commonly used by institutions. By facilitating interoperability, BMCC TRMS can leverage existing data and workflows, enhancing efficiency and reducing duplication of effort.

By incorporating these scalability considerations into its design and development, BMCC TRMS can evolve and grow alongside the changing needs of educators and educational institutions, ensuring its continued relevance and effectiveness in the long term.

Limitations

1.**Initial Implementation Complexity**: Setting up BMCC TRMS may require considerable time and resources for data migration, system configuration, and user training. Institutions with limited technical expertise or resources may face challenges during the initial implementation phase.

2.**User Adoption Challenges**: Encouraging educators to adopt new technology like BMCC TRMS can be difficult, especially if they are accustomed to traditional methods. Resistance to change or lack of familiarity with digital tools may hinder user adoption and utilization.

Assumption

1. **Internet Connectivity**: Users are assumed to have access to the internet to use this platform.

Technologies Used

Programming Language

- MongoDB, Express, React, and Node. Js(MERN)

Styling Language

- Cascading Style Sheets

External Libraries Used (Frontend)

- axios: ^1.6.5
- bootstrap: ^5.3.2
- jquery: ^3.7.1
- jspdf: ^2.5.1
- jspdf-autotable: ^3.8.2
- nanoid: ^5.0.4
- popper.js: ^1.16.1
- react: ^18.2.0
- react-dom: ^18.2.0
- react-icons: ^4.12.0
- react-router-dom: ^6.21.1
- socket.io-client: ^4.7.4

External Libraries Used (Backend)

- bcrypt: ^5.1.1
- cors: ^2.8.5
- express: ^4.18.2
- mongoose: ^8.0.3
- nodemailer: ^6.9.13
- socket.io: ^4.7.4

Collection Schema Definition

- **Cyrus Hall**

```
const cyrusSchema = mongoose.Schema({
  date: {
    type: String,
    required: true
  },
  bookedBy: {
    type: String,
    required: true
  },
  email:{
    type : String,
    required : true
  },
  startTime: {
    type: Date,
    required: true
  },
  endTime: {
    type: Date,
    required: true
  },
purpose: {
    type: String,
    required: true
  }
})
```

- **Tata Hall**

```
const tataSchema = mongoose.Schema({
  date: {
    type: String,
    required: true
  },
  bookedBy: {
    type: String,
    required: true
  },
  email:{
    type : String,
    required : true
  },
  startTime: {
    type: Date,
    required: true
  },
  endTime: {
    type: Date,
    required: true
  },
  purpose: {
    type: String,
    required: true
  }
})
```

- **User**

```
const userSchema = mongoose.Schema({
  email : {
    type : String,
    required : true,
    unique : true
  },
  password: {
    type: String,
    required: true
  },
 name: {
    type: String,
    required: true
  },
  userType : {
    type : String,
    required : true
  }
})
```

- **Time Slot**

```
const timeSlotSchema = {
  bookedBy: {
    type: String,
    required: true
  },
  slot: {
    type: Number,
    required: true
  },
  subject : {
    type : String,
    required : true
  },
  email : {
    type : String,
    required : true
  }
}
```

- **Lab**

```
const labSchema = mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  day: {
    type: String,
    required: true
  },
  timeSlots: [timeSlotSchema]
})
```

- **Timetable**

```
const timetableSchema = new mongoose.Schema({
 class: String,
 timetable: Array,
});
```

- **File**

```
const fileSchema = new mongoose.Schema({
  filename: String,
  contentType: String,
  size: Number,
  filePath: String
});
```

- **Students**

```
const Students_TY_BCASchema = new mongoose.Schema({
     rollNumber: String,
     studentName: String,
     class:String
   });
```

Types of Users and their Functions

1. Teacher (Primary User)

- My Tasks (Add, Complete, Delete, Undo)
- Lab (Book, View, Delete)
- Hall (Book, View, Delete)
- Timetable (View)
- Group List Generator
- Files (Upload, View, Download, Delete)

2. Admin

- Add New User
- Delete User
- Lab (View)
- Hall (View)
- Timetable (Create)