

MERN Mavericks: Code, Create And Conquer! – 2025

Smart Campus Portal Using MERN Stack

E-Vidya Saathi

Smart Campus Portal Using MERN Stack

Presented by Akanksha (B.Tech ECE, GITAM Visakhapatnam)

1. Project Objectives

- Develop a centralized Smart Campus Portal for managing student activities, faculty interactions, event notifications, and academic placement tracking.
- Provide separate dashboards for Students, Faculty, and Admins to streamline campus operations.
- Digitize and optimize campus activities for efficiency, transparency, and accessibility.

2. Technology Stack Used

Frontend:

- React.js
- CSS3, Bootstrap
- HTML5

Backend:

- Node.js
- Express.js
- MongoDB (MongoDB Atlas)
- JWT

Additional Libraries:

- jQuery (if used)
- Various npm packages (see package.json)

3. Codebase Architecture & File Structure

High-level Structure:

```
source-code/
├── client/           # Frontend (React)
│   ├── public/
│   ├── src/
│   ├── package.json
│   └── README.md
└── server/          # Backend (Node.js/Express)
    ├── config/
    ├── controllers/
    ├── middleware/
    ├── models/
    ├── routes/
    ├── utils/
    ├── server.js
    └── package.json
```

4. Features and Their Descriptions

Student Dashboard:

- Access to courses
- Attendance tracking
- Grade tracking
- Event notifications

Faculty Dashboard:

- Course management
- Assignment management
- Performance insights

Admin Panel:

- User management
- Event monitoring
- Campus-wide alerts

Event Management:

- Upcoming events
- Event registration
- Event notifications

Placement Tracking:

- Placement records
- Admin dashboard

Authentication & Security:

- JWT-based authentication

- Role-based access
- Error handling

Responsive Design:

- Mobile, tablet, and desktop support
- Smooth navigation

5. Database Schema Diagram

Collections: Users, Courses, Events, Placements, Attendance, Notifications

Relationships: One-to-many and Many-to-many (as described)

6. Installation Guide

Frontend:

```
cd source-code/client
npm install
npm start
```

Backend:

```
cd source-code/server
npm install
# Add your .env file with MongoDB URI and JWT secret
npm start
```

7. API Documentation

Endpoints:

```
POST /api/auth/login - User login
GET /api/courses - List all courses
POST /api/events - Create new event (admin/faculty)
GET /api/placements - Get placement records
```

8. Feature List (Summary)

- Student Dashboard
- Faculty Dashboard
- Admin Panel
- Event Management
- Placement Tracking
- JWT Authentication and role-based access
- Responsive UI
- Error handling and validation

9. Implementation Details

Frontend: React.js and Bootstrap

Backend: Node.js, Express.js with MongoDB Atlas

Authentication: JWT and role-based middleware

Deployment: (Add cloud info if hosted)

10. Challenges Faced

- Designing comprehensive schema
- Secure authentication
- Real-time event notifications

11. Future Scope

- Real-time chat
- Company dashboards in placement module
- Admin analytics
- React Native mobile app

12. Code Quality

- Follows modular design
- Well-commented
- Clean commit history

Frontend: <https://github.com/Akanksha9821/E-Vidya-Saathi/tree/main/source-code/client>

Backend: <https://github.com/Akanksha9821/E-Vidya-Saathi/tree/main/source-code/server>