

Full Stack Development with MERN

Database Design and Development Report

Date	17.04.24
Team ID	PNT2022TMIDSWTID1721142115
Project Name	Video Conferencing Application
Maximum Marks	10 Marks

Project Title: Video Conferencing Application (Smart Meet)

Date: 17.07.24

Prepared by: MeetMinds(SWTID1721142115)

Objective

The objective of this report is to outline the database design and implementation details for the Video Conferencing Application project, including schema design and database management system (DBMS) integration.

Technologies Used

- **Database Management System (DBMS):** MongoDB
- **Object-Document Mapper (ODM):** Mongoose

Design the Database Schema

The database schema is designed to accommodate the following entities and relationships:

1. Users

- Attributes: username, email, password

2. Rooms

- Attributes: roomName, host, meetType, meetDate,meetTime, participants, currentParticipants

Implement the Database using MongoDB

The MongoDB database is implemented with the following collections and structures:

Database Name: Video Conferencing

1. Collection: users

- Schema:

...

username:{

 type: String,

 require: true

},

email:{

 type: String,

 require: true,

 unique: true

},

password:{

 type: String,

 require: true

}

...

2. Collection: Rooms

- Schema:

...

roomName:{

 type: String

},

host: {

 type: String,

 require: true

},

meetType:{

 type: String,

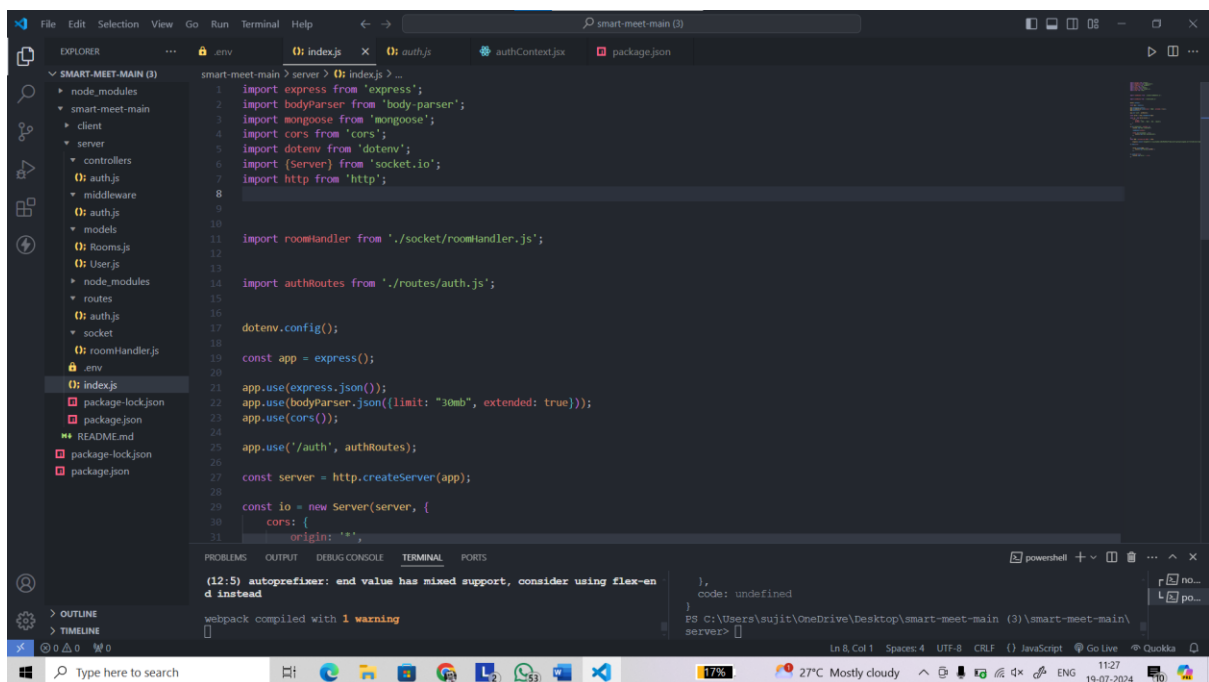
```

},
meetDate:{
  type: String,
},
meetTime:{
  type: String,
},
participants: {
  type: Array
},
currentParticipants: {
  type: Array
}
'''

```

Integration with Backend

- Database connection



The screenshot shows a Visual Studio Code editor window with a project named 'smart-meet-main'. The Explorer sidebar on the left shows the file structure, including 'node_modules', 'smart-meet-main', 'client', 'server', 'controllers', 'auth.js', 'middleware', 'models', 'Rooms.js', 'User.js', 'node_modules', 'routes', 'socket', 'roomHandler.js', '.env', 'index.js', 'package-lock.json', 'package.json', 'README.md', and another 'package-lock.json'. The main editor area displays the 'index.js' file, which contains the following code:

```
const io = new Server(server, {
  cors: {
  }
});

io.on("connection", (socket) =>{
  console.log("User connected");

  roomHandler(socket);

  socket.on("disconnect", ()=>{
    console.log("user disconnected");
  })
})

const PORT = process.env.PORT || 6002;

mongoose.connect('mongodb+srv://sujithy2021:zdhLPHoKVSAxYr3@cluster0.gkroay6.mongodb.net/?retryWrites=true&majority=&appName=Cluster0', {
}).then(()=>{

  server.listen(PORT, ()=>{
    console.log(`Running @${PORT}`);
  });

}).catch((err)=>{
  console.log("Error: ", err);
})
```

The bottom panel of the editor shows the 'TERMINAL' tab with the following output:

```
(12:5) autoprefixer: end value has mixed support, consider using flex-en
d instead

webpack compiled with 1 warning

},
code: undefined
}
PS C:\Users\sujit\OneDrive\Desktop\smart-meet-main (3)\smart-meet-main\
server>
```

The status bar at the bottom indicates the file is 'Ln 8, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'JavaScript', and 'Go Live' is active. The system tray shows the date and time as '11:27 19-07-2024'.

- The backend APIs interact with MongoDB using Mongoose ODM Key interactions include:
 - User Management: CRUD operations for users.
 - Room Management: CRUD operations for Rooms, with user authentication.