

Project Development Phase

Model Performance Testing

Date	20 Feb 2026
Team ID	LTVIP2026TMIDS79402
Project Name	Booknest : Where stories nestle

Model Performance Testing:

S.NO.	Parameter	Values
1.	System Architecture	Frontend: React.js
	Backend: Node.js + Express.js	
	Database: MongoDB	
	Authentication: JWT + bcrypt	
2.	API Response Time	Average Response Time: < 300 ms (local testing)
	Page Load Time: < 3 seconds	
3.	Authentication Performance	Login & Token Generation: < 500 ms
	Password Encryption: bcrypt with secure hashing	
4.	Database Performance	Efficient CRUD operations using Mongoose
	Indexed queries for faster book search	
5.	Load Handling	Tested with multiple concurrent requests using Postman
	System handles simultaneous users without crashing	
6.	Frontend Performance	Fast rendering using React virtual DOM
	Dynamic state updates without full page reload	

7.	Optimization Applied	Environment variables for secure config
	Proper error handling & validation	
	Modular folder structure for maintainability	

Screenshots:

The screenshots are given below

```

const PORT = process.env.PORT || 6001;
mongoose.connect(process.env.MONGO_URL, {
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(()=>{

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

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

```

The screenshot shows a code editor window with a dark theme. The file is named "User.js". The code defines a MongoDB schema for a "User" model. It includes fields for "username", "email", and "password", each with a type of "String" and a required field. The "email" field also has a unique constraint. The schema is then used to create a "User" model.

```
JS User.js X
server > models > JS User.js > ...
1 import mongoose from 'mongoose';
2
3 const UserSchema = new mongoose.Schema({
4   username:{
5     type: String,
6     require: true
7   },
8   email:{
9     type: String,
10    require: true,
11    unique: true
12 },
13   password:{
14     type: String,
15     require: true
16 },
17 });
18
19 const User = mongoose.model("users", UserSchema);
20 export default User;
```

The above are some screenshots of my project.