

**COSC 310**

# **Maintenance Manual**

Khizar Aamir, Abdullah Naqvi, Abdur Rehman, Haider Ali, Abdullah Munir

This manual provides essential information for maintaining the Task Management System, focusing on critical operational aspects.

## **Database Management**

### **Database Configuration**

- **Database Type:** MongoDB (hosted on MongoDB Atlas)
- **Connection:** The application connects to MongoDB using the connection string stored in the .env file
- **Environment Variable:** MONGO\_URI in backend/.env

### **Database Backup Recommendations**

#### ***1. Regular Backups:***

- Use MongoDB Atlas automated backups if using their hosted solution
- For self-hosted MongoDB, implement scheduled mongodump operations

#### ***2. Backup Verification:***

- Periodically test backup restoration to ensure data integrity
- Store backups securely with appropriate access controls

### **Database Restoration**

In case of data loss, restore from backup using MongoDB's standard procedures or Atlas dashboard if using their service.

## **Environment Variables**

The application relies on the following environment variables defined in backend/.env:

MONGO\_URI=<MongoDB connection string>  
PORT=<Application port, defaults to 3200>  
URL=<Frontend URL>

BREVO\_USER=<Brevo SMTP username>  
BREVO\_PASS=<Brevo SMTP password>  
BREVO\_SENDER=<Email sender address>  
REFRESH\_TOKEN\_SECRET=<JWT refresh token secret>  
ACCESS\_TOKEN\_SECRET=<JWT access token secret>

**Important:** Never commit these values to version control. Properly secure the .env file with restricted access permissions.

## **Email Service**

The application uses Brevo for sending emails:

- **Implementation:** Found in backend/utils/emailTransporter.js
- **Features:** Uses Handlebars templates for email formatting
- **Credentials:** Stored in environment variables (BREVO\_USER, BREVO\_PASS, BREVO\_SENDER)

## **Logging System**

### **Logging Configuration**

- **Logger:** Winston (configured in backend/logs/logger.js)
- **Log Levels:** error, warn, info, debug, success
- **Log Files:** Errors are written to backend/logs/errors.log

### **Monitoring Logs**

- Regularly review error logs to identify and address issues
- Consider implementing a log aggregation solution for production environments

## **Authentication Security**

- **JWT Implementation:** The system uses JSON Web Tokens for authentication

- **Token Management:** Refresh and access tokens are implemented for secure sessions
- **Security Keys:** JWT secrets are stored in environment variables
- **Token Verification:** Implemented in backend/middleware/verifyJWT.js

## **Server Hosting**

- **Port Configuration:** The backend runs on the port specified in the PORT environment variable (default: 3200)
- **Socket.io:** Real-time features are implemented using Socket.io with authenticated connections

## **Maintenance Checklist**

Task	Frequency	Description
Check error logs	Weekly	Review error logs for issues requiring attention
Update dependencies	Monthly	Update npm packages to maintain security
Database backup	According to data importance	Ensure backups are functioning properly
JWT secret rotation	Quarterly	Change JWT secrets for enhanced security
Email service verification	Monthly	Ensure email service is functioning correctly

## **Troubleshooting Common Issues**

### **Database Connection Issues**

- Verify MongoDB connection string in .env file
- Check MongoDB Atlas dashboard for service status
- Review backend/logs/errors.log for specific connection errors

### **Email Sending Failures**

- Verify Brevo credentials in .env file
- Check email service logs in backend/logs/errors.log
- Test SMTP connection manually to confirm service availability

### **Authentication Problems**

- Ensure JWT secrets are properly set in environment variables
- Check token expiration settings in authentication controller
- Verify frontend is correctly storing and using tokens