MEAN Stack Application Support Document

1. Introduction

Application Name: MyMeanApp

Version: 1.0.0

Description: A web application built using the MEAN stack (MongoDB, Express.js, Angular, Node.js) for task management and collaboration.

Support Contact:

- **Email:** support@mymeanapp.com
- **Phone:** +1-555-345-6789
- **Support Hours:** Monday to Friday, 9 AM - 5 PM (PST)

---

2. Common Issues and Troubleshooting

2.1. Frontend Issues

Issue: Application Not Loading

- **Description:** The Angular frontend does not load or displays a blank page.
- **Troubleshooting Steps:**
  1. Check the browser console for JavaScript errors.
  2. Verify that the frontend assets are correctly built and served.
  3. Ensure that the API endpoints are correctly configured in the Angular environment settings.
  4. Confirm that the Angular app is correctly deployed and the Nginx server is running.

Issue: API Requests Fail

- **Description:** The Angular application fails to fetch data from the backend API.

- **Troubleshooting Steps:**
  1. Verify that the backend API is running and accessible.
  2. Check the network tab in the browser's developer tools for failed requests and error messages.
  3. Ensure that CORS (Cross-Origin Resource Sharing) is correctly configured on the backend server.

2.2. Backend Issues

Issue: API Server Crashes
- **Description:** The Node.js server crashes or is unresponsive.
- **Troubleshooting Steps:**
  1. Check the Node.js application logs for error messages: `tail -f /var/log/mymeanapp/node.log`.
  2. Verify that all required environment variables are correctly set.
  3. Ensure that the Node.js service is running: `sudo systemctl status mymeanapp.service`.
  4. Restart the Node.js service if needed: `sudo systemctl restart mymeanapp.service`.

Issue: Performance Degradation
- **Description:** The backend API experiences slow response times or high latency.
- **Troubleshooting Steps:**
  1. Monitor server performance metrics (CPU, memory, disk I/O).
  2. Check for database performance issues and optimize queries.
  3. Analyze and optimize Node.js code and middleware.
  4. Review Nginx configuration for potential bottlenecks.

2.3. Database Issues

Issue: Database Connection Errors
- **Description:** The application cannot connect to the MongoDB database.
- **Troubleshooting Steps:**
  1. Verify that MongoDB is running and accessible: `sudo systemctl status mongod`.
  2. Check the MongoDB connection string and credentials in the application configuration.
  3. Review MongoDB logs for any connection or authentication errors.

4. Ensure network connectivity between the application server and MongoDB.

Issue: Data Inconsistencies

- **Description:** The data in the database appears to be incorrect or inconsistent.
- **Troubleshooting Steps:**
  1. Check for recent changes or migrations that may have affected the data.
  2. Review the application's data handling and validation logic.
  3. Verify the integrity of the database schema and indexes.
  4. Perform data consistency checks and repairs if necessary.

---

3. Support Procedures

3.1. Incident Reporting

Procedure:

1. **Log the Incident:** Use the internal ticketing system to log details of the incident, including the date, time, and a description of the issue.
2. **Assign Priority:** Determine the priority level based on the impact on users (e.g., Critical, High, Medium, Low).
3. **Assign to Team:** Forward the ticket to the appropriate support or development team for resolution.
4. **Track Progress:** Monitor the progress of the incident resolution and update the ticket with any relevant information.

3.2. Escalation Procedures

Procedure:

1. **Identify Need for Escalation:** If the issue cannot be resolved within the standard support timeframe or requires specialized knowledge, identify the need for escalation.

2. **Escalate to Higher-Level Support:** Forward the incident to the senior support team or subject matter experts.

3. **Provide Detailed Information:** Ensure all relevant information is included in the escalation to assist in a quick resolution.

4. **Follow Up:** Monitor the escalation and provide updates to the affected stakeholders.

3.3. Maintenance Procedures

Procedure:

1. **Schedule Maintenance:** Plan and schedule maintenance windows with minimal impact on users.

2. **Notify Users:** Inform users of upcoming maintenance and potential downtime.

3. **Perform Maintenance:** Execute the maintenance tasks, including updates, backups, and performance optimizations.

4. **Verify Completion:** Ensure that all maintenance tasks are completed successfully and that the application is functioning as expected.

5. **Update Documentation:** Document any changes made during the maintenance and update relevant system documentation.

---

4. Contact Information

Support Team:

- **Lead Support Specialist:** Alice Johnson
  Email: alice.johnson@mymeanapp.com
  Phone: +1-555-678-9101

- **Backup Support Specialist:** Bob Williams
  Email: bob.williams@mymeanapp.com
  Phone: +1-555-789-0123

Development Team:

- **Lead Developer:** Jane Smith
  Email: jane.smith@mymeanapp.com
  Phone: +1-555-234-5678

- **Backend Developer:** John Doe
  Email: john.doe@mymeanapp.com
  Phone: +1-555-123-4567

---

5. Documentation and Resources

- **User Guide:** [User Guide Link](https://docs.mymeanapp.com/user-guide)
- **API Documentation:** [API Docs Link](https://docs.mymeanapp.com/api)
- **Developer Documentation:** [Developer Docs Link](https://docs.mymeanapp.com/developer)

End of Document
