Experiment 7.1 Middleware Implementation for Logging and Bearer Token Authentication

Aim: To implement middleware in Express.js for logging and Bearer Token-based authentication.

Theory:

1. Middleware functions in Express.js are used to process requests before they reach the route handlers.
2. Logging middleware helps record important request and response details such as method, URL, and status.
3. Authentication middleware ensures that only authorized users can access protected endpoints.
4. Bearer tokens are used in the Authorization header to securely transmit user credentials.
5. JWT (JSON Web Token) is commonly used for token-based authentication, providing integrity and expiration control.
6. Combining logging and authentication middleware enhances both security and traceability of an application.

Code Implementation:

```
const express=require('express');
const jwt=require('jsonwebtoken');
const bodyParser=require('body-parser');

const SECRET='your-very-secure-secret';

function logger(req,res,next){
  const start=Date.now();
  res.on('finish',()=>{
    const duration=Date.now()-start;
    console.log(`${req.ip} ${req.method} ${req.originalUrl} ${res.statusCode} ${duration}ms`);
  });
  next();
}

function authenticateToken(req,res,next){
  const auth=req.headers['authorization'];
  if(!auth||!auth.startsWith('Bearer ')) return res.status(401).json({error:'Unauthorized'});
  const token=auth.slice(7);
  jwt.verify(token,SECRET,(err,decoded)=>{
    if(err) return res.status(403).json({error:'Forbidden'});
    req.user=decoded;
    next();
  });
}

const app=express();
app.use(bodyParser.json());
app.use(logger);

app.get('/public',(req,res)=>res.json({message:'public endpoint'}));
app.get('/private',authenticateToken,(req,res)=>res.json({message:'private',user:req.user}));

app.post('/login',(req,res)=>{
  const {username}=req.body;
  if(!username) return res.status(400).json({error:'Username required'});
  const token=jwt.sign({username},SECRET,{expiresIn:'1h'});
  res.json({token});
});
const port=process.env.PORT||3000;
```

```
app.listen(port,()=>console.log(`Server running on port ${port}`));
```