BÁO CÁO: XÂY D■NG SERVER V■I NODE.JS + EXPRESS.JS

1. Cài ■■t môi tr■■ng

- Cài ■■t Node.js (t■ https://nodejs.org/)
- Ki■m tra b■ng l■nh:

```
    node -v
    npm -v
```

2. T■o d■ án m■i

```
mkdir forum-server
cd forum-server
npm init -y
```

3. Cài ■■t các gói c■n thi■t

```
npm install express mysql2 dotenv cors body-parser
```

4. T■o c■u trúc th■ m■c

```
forum-server/
■■■ routes/
■   ■■■ auth.js
■■■ controllers/
■   ■■■ authController.js
■■■ config/
■   ■■■ db.js
■■■ .env
■■■ server.js
■■■ package.json
```

5. File c■u hình môi tr■■ng `.env`

```
PORT=3000
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=yourpassword
DB_NAME=forum
```

6. K■t n■i MySQL – config/db.js

```
const mysql = require('mysql2');
require('dotenv').config();

const db = mysql.createConnection({
  host: process.env.DB_HOST,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME
});

db.connect((err) => {
  if (err) throw err;
  console.log('K■t n■i MySQL thành công');
});

module.exports = db;
```

7. API ■■n gi■n – routes/auth.js

```
const express = require('express');
const router = express.Router();
const authController = require('../controllers/authController');
```

```
router.post('/register', authController.register);
router.post('/login', authController.login);

module.exports = router;
```

8. Logic x■ lý – controllers/authController.js

```
const db = require('../config/db');

exports.register = (req, res) => {
  const { username, password } = req.body;
  const sql = "INSERT INTO users (username, password) VALUES (?, ?)";
  db.query(sql, [username, password], (err, result) => {
    if (err) return res.status(500).json({ error: err });
    res.status(201).json({ message: '■■ng ký thành công' });
  });
};

exports.login = (req, res) => {
  const { username, password } = req.body;
  const sql = "SELECT * FROM users WHERE username = ? AND password = ?";
  db.query(sql, [username, password], (err, results) => {
    if (err) return res.status(500).json({ error: err });
    if (results.length > 0) {
      res.json({ message: '■■ng nh■p thành công' });
    } else {
      res.status(401).json({ message: 'Sai tài kho■n ho■c m■t kh■u' });
    }
  });
};
```

9. Kh■i ch■y server – server.js

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
require('dotenv').config();

const app = express();
const PORT = process.env.PORT || 3000;

const authRoutes = require('./routes/auth');

app.use(cors());
app.use(bodyParser.json());

app.use('/api/auth', authRoutes);

app.listen(PORT, () => {
  console.log(`Server ■ang ch■y t■i http://localhost:${PORT}`);
});
```

10. Ki■m tra

S■ d■ng Postman ho■c frontend ■■ ki■m tra:

POST http://localhost:3000/api/auth/register

```
POST http://localhost:3000/api/auth/login
```