

server.js

```
require("dotenv").config();

const express = require("express");

const app = express();

const port = process.env.PORT;

app.use(express.json());
app.use(express.urlencoded({ extended: true }));

app.get("/", (req, res) => {
  res.send(
    `Hello World! This is the server running on port ${port}. This is a static page.`
  );
});

app.use("/api/user", require("./routes/userRoutes"));
app.use("/api/admin", require("./routes/adminRoutes"));

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

The image shows a web development IDE interface. At the top, there's a menu bar with 'File', 'Edit', 'Selection', 'View', 'Go', 'Terminal', 'Help', and a search icon. Below the menu is a toolbar with various icons for file operations and development tools. The main workspace is divided into three panels. The left panel is the 'Explorer' showing a file tree with folders like 'Backend', 'config', 'controllers', 'middlewares', 'node_modules', 'routes', 'adminRoutes.js', 'userRoutes.js', 'utils', '.env', '.gitignore', 'package-lock.json', 'package.json', 'README.MD', and 'Frontend'. The middle panel is a 'Webview' showing a static page with the text 'Hello World! This is the server running on port 8000. This is a static page.' The right panel is the 'Terminal' showing the output of a Node.js server running on port 8000. The terminal text includes: '[nodemon] 3.1.9', '[nodemon] to restart at any time, enter `rs`', '[nodemon] watching path(s): *.*', '[nodemon] watching extensions: js,mjs,cjs,json', '[nodemon] starting `node server.js`', 'Database Connection Successful', and 'Server is running on port 8000'.