|  |
| --- |
| import jwt from "jsonwebtoken";  const loginUser = async (req, res) => {  const { email, password } = req.body;  const token = req.headers.authorization?.split(" ")[1]; // Extract token from Authorization header  try {  // Step 1: Verify existing token if provided  if (token) {  try {  const decoded = jwt.verify(token, process.env.JWT\_SECRET);  return res.json({ success: true, message: "Already authenticated", token });  } catch (err) {  // Token invalid or expired; proceed with normal login  }  }  // Step 2: Check user credentials  const user = await userModel.findOne({ email });  if (!user) {  return res.json({ success: false, message: "User does not exist" });  }  const isMatch = await bcrypt.compare(password, user.password);  if (!isMatch) {  return res.json({ success: false, message: "Invalid credentials" });  }  // Step 3: Generate a new token  const newToken = createToken(user.\_id);  res.json({ success: true, token: newToken });  } catch (error) {  console.log(error);  res.json({ success: false, message: "Error" });  }  }; |
|  |
|  |
|  |