Assignment_3b\controllers\userController.js

```
const User = require('../models/User');

// @desc    Register user
// @route   POST /api/users/register
// @access  Public
exports.registerUser = async (req, res) => {
  try {
    const { name, email, password } = req.body;

    // Check if user exists
    const userExists = await User.findOne({ email });

    if (userExists) {
      return res.status(400).json({
        success: false,
        message: 'User already exists',
      });
    }

    // Create user
    const user = await User.create({
      name,
      email,
      password,
    });

    // Generate token
    const token = user.getSignedJwtToken();

    res.status(201).json({
      success: true,
      token,
      user: {
        id: user._id,
        name: user.name,
        email: user.email,
      },
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: 'Server Error',
      error: error.message,
    });
  }
};

// @desc    Login user
// @route   POST /api/users/login
// @access  Public
exports.loginUser = async (req, res) => {
  try {
    const { email, password } = req.body;

    // Validate email & password
    if (!email || !password) {
      return res.status(400).json({
        success: false,
        message: 'Please provide an email and password',
      });
    }

    // Check for user
    const user = await User.findOne({ email }).select('+password');

    if (!user) {
      return res.status(401).json({
        success: false,
        message: 'Invalid credentials',
      });
    }

    // Check if password matches
    const isMatch = await user.matchPassword(password);

    if (!isMatch) {
      return res.status(401).json({
        success: false,
        message: 'Invalid credentials',
      });
    }

    // Generate token
    const token = user.getSignedJwtToken();

    res.status(200).json({
      success: true,
      token,
      user: {
        id: user._id,
        name: user.name,
        email: user.email,
      },
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: 'Server Error',
      error: error.message,
    });
  }
};

// @desc    Get user data
// @route   GET /api/users/me
// @access  Private
exports.getUserProfile = async (req, res) => {
  try {
    const user = await User.findById(req.user.id);

    res.status(200).json({
      success: true,
      user: {
        id: user._id,
        name: user.name,
        email: user.email,
        createdAt: user.createdAt,
      },
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: 'Server Error',
      error: error.message,
    });
  }
};

// @desc    Update user profile
// @route   PUT /api/users/me
// @access  Private
exports.updateUserProfile = async (req, res) => {
  try {
    const updatedUser = await User.findByIdAndUpdate(
      req.user.id,
      { $set: req.body },
      { new: true, runValidators: true }
    );

    res.status(200).json({
      success: true,
      user: {
        id: updatedUser._id,
        name: updatedUser.name,
        email: updatedUser.email,
        createdAt: updatedUser.createdAt,
      },
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: 'Server Error',
      error: error.message,
    });
  }
};
```