```
const express = require('express');
const router = express.Router();
const Cost = require('../models/costs');
const User = require('../models/users');

/**
 * @route POST /api/add
 * @desc Adds a new cost entry for a user
 * @access Public
 */
router.post('/add', async (req, res) => {
    try {
        const { userid, category, description, sum, date } = req.body;

        // Check if all required fields are provided
        if (!userid || !category || !description || !sum) {
            return res.status(400).json({ error: 'Missing required fields: userid, category, description, sum'
        }

        // If date is not provided, use the current date and time
        const costDate = date ? new Date(date) : new Date();

        // Create a new cost item
        const cost = new Cost({
            userid,
            category,
            description,
            sum,
            date: costDate
        });

        // Save the new cost item to the database
        const savedCost = await cost.save();

        // Respond with the newly added cost item
        res.status(201).json(savedCost);
    } catch (error) {
        // Catch any errors and return them
        res.status(500).json({ error: error.message });
    }
});

/**
 * @route GET /api/report
 * @desc Fetches monthly report for a specific user
 * @access Public
 */
router.get('/report', async (req, res) => {
    const { id: userid, year, month } = req.query;

    // Validate input parameters
    if (!userid || !year || !month) {
        return res.status(400).json({ error: 'Missing required query parameters: id, year, month' });
    }
```

```
    try {
        const startOfMonth = new Date(year, month - 1, 1); // Start of month
        const endOfMonth = new Date(year, month, 0);       // End of month

        // Fetch the costs within the month and year range
        const costs = await Cost.find({
            userid: userid,
            date: { $gte: startOfMonth, $lte: endOfMonth }
        });

        if (costs.length === 0) {
            return res.status(404).json({ error: 'No costs found for this user in the specified month and year'
        }

        // Prepare the response structure
        const categories = {
            food: [],
            education: [],
            health: [],
            housing: []
        };

        // Group costs by category
        costs.forEach(cost => {
            const { category, sum, description, date } = cost;
            const day = new Date(date).getDate(); // Extract the day

            if (categories[category]) {
                categories[category].push({ sum, description, day });
            }
        });

        // Format the final report
        const report = {
            userid: userid,
            year: parseInt(year),
            month: parseInt(month),
            costs: Object.keys(categories).map(category => ({ [category]: categories[category] }))
        };

        res.json(report);
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});

/**
 * @route GET /api/users/:id
 * @desc Retrieves user details and their total spending
 * @access Public
 */
router.get('/users/:id', async (req, res) => {
    const userId = req.params.id;

    try {
        const user = await User.findOne({ id: userId });
        if (!user) {
            return res.status(404).json({ error: 'User not found' });
        }
```

```
        // Calculate total costs
        const totalCosts = await Cost.aggregate([
            { $match: { userid: userId } },
            { $group: { _id: null, total: { $sum: '$sum' } } },
        ]);

        res.json({
            first_name: user.first_name,
            last_name: user.last_name,
            id: user.id,
            marital_status: user.marital_status,
            total: totalCosts[0]?.total || 0,
        });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});

/**
 * @route GET /api/about
 * @desc Provides information about the development team
 * @access Public
 */
router.get('/about', async (req, res) => {
    const team = [
        { first_name: 'Omri', last_name: 'Shmuel'},
        { first_name: 'Omer', last_name: 'Tzaadi' },
        { first_name: 'Astewuol', last_name: 'Tsagaow' },
    ];

    res.json(team);
});

module.exports = router;
```
