```solidity

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract WeeklyAdminPayments {

address public owner;

address payable public adminWallet; // Admin's blockchain wallet

uint256 public paymentAmount;

uint256 public lastPaymentTime;

event PaymentSent(address indexed admin, uint256 amount);

constructor(address payable \_adminWallet, uint256 \_paymentAmount) {

owner = msg.sender;

adminWallet = \_adminWallet;

paymentAmount = \_paymentAmount;

lastPaymentTime = block.timestamp; // Initialize payment schedule

}

modifier onlyOwner() {

require(msg.sender == owner, "Only owner can perform this action");

\_;

}

function setAdminWallet(address payable \_adminWallet) public onlyOwner {

adminWallet = \_adminWallet;

}

function setPaymentAmount(uint256 \_paymentAmount) public onlyOwner {

paymentAmount = \_paymentAmount;

}

function payAdmin() public onlyOwner {

require(block.timestamp >= lastPaymentTime + 7 days, "Weekly payment not due yet");

require(address(this).balance >= paymentAmount, "Not enough funds in the contract");

adminWallet.transfer(paymentAmount);

lastPaymentTime = block.timestamp;

emit PaymentSent(adminWallet, paymentAmount);

}

function depositFunds() public payable {

// Deposit funds into the contract

}

function getContractBalance() public view returns (uint256) {

return address(this).balance;

}

}

```