

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μάθημα Προπτυχιακών Σπουδών:
Τεχνολογίες Blockchain και Εφαρμογές

Ακαδημαϊκό Έτος:
2023-2024

Εξάμηνο 8
Εργασία

Ονοματεπώνυμο	Δημήτρης Στυλιανού
Αριθμός Μητρώου	Π20004
Υπεύθυνη Καθηγήτρια	Αριστέα Κοντογιάννη

Ημερομηνία παράδοσης:
11 Ιουλίου 2024

Περιεχόμενα

1	Ανάλυση Κώδικα	2
1.1	Δομή δεδομένων Poll	2
1.2	Συνάρτηση createNewPoll	3
1.3	Συνάρτηση castVote	4
1.4	Συνάρτηση closePoll	5
1.5	Συνάρτηση showPollVotes	5
2	Στιγμιότυπα Οθόνης	6
3	Παράδειγμα Εκτέλεσης	7

1 Ανάλυση Κώδικα

Ο πηγαίος κώδικας της εργασίας βρίσκεται στο αρχείο **smartVote.sol**.

1.1 Δομή δεδομένων Poll

Η αναπαράσταση μιας ψηφοφορίας έγινε με μια δομή δεδομένων (struct) με όνομα **"Poll"**, η οποία περιέχει τους εξής τύπους δεδομένων:

- **string question**, αντιστοιχεί στην ερώτηση της ψηφοφορίας
- **address creatorAddress**, αντιστοιχεί στην διεύθυνση λογαριασμού του δημιουργού της ψηφοφορίας
- **bool isOpen**, υποδεικνύει εάν η ψηφοφορία είναι ανοικτή
- **mapping(address => bool) votes**, διατηρεί τις ψήφους της ψηφοφορίας (αποθηκεύοντας ως κλειδί την διεύθυνση λογαριασμού του ψηφοφόρου και την απάντηση του)
- **uint numOfPositiveVotes**, αποθηκεύει το πλήθος θετικών ψήφων
- **uint numOfNegativeVotes**, αποθηκεύει το πλήθος αρνητικών ψήφων

Για την αποθήκευση κάθε ψηφοφορίας δημιουργήθηκε ένα πίνακας με όνομα **"polls"**.

```
/**
 * @notice Represents a poll with a question, creator's address, open status, vote mapping, and positive/negative vote counters.
 */
struct Poll {
    string question;           // The question for the poll
    address creatorAddress;    // The address of the poll creator
    bool isOpen;               // The status indicating if the poll is open for voting
    mapping(address => bool) votes; // Mapping to track votes (true for voted, false for not voted)
    uint numOfPositiveVotes;   // The number of positive (true) votes received
    uint numOfNegativeVotes;   // The number of negative (false) votes received
}

Poll[] private polls; // Array to store all the polls
```

Εικόνα 1: Δομή δεδομένων Poll

1.2 Συνάρτηση createNewPoll

Για την δημιουργία μιας νέας ψηφοφορίας υλοποιήθηκε η συνάρτηση **createNewPoll**, η οποία δέχεται ως παράμετρο την ερώτηση της ψηφοφορίας και δημιουργεί μια μεταβλητή τύπου Poll, στην οποία εκχωρεί:

1. την δοθείσα ερώτηση
2. την διεύθυνση λογαριασμού του δημιουργού της ψηφοφορίας
3. την τιμή true υποδεικνύοντας ότι η ψηφοφορία είναι ανοικτή

Τέλος, η μεταβλητή τύπου Poll αποθηκεύεται στην λίστα polls.

```
/**
 * @notice Creates a new poll with the given question.
 * @dev Adds a new Poll struct to the polls array.
 * @param question The question for the new poll.
 */
function createNewPoll(string calldata question) external {
    Poll storage poll = polls.push(); // Add a new Poll and get a reference to it
    poll.question = question;          // Set the poll's question to the given question
    poll.creatorAddress = msg.sender;   // Set creatorAddress to the caller's address
    poll.isOpen = true;                 // Set isOpen to true
}
```

Εικόνα 2: Συνάρτηση createNewPoll

1.3 Συνάρτηση castVote

Για την καταγραφή των ψήφων των χρηστών, υλοποιήθηκε η συνάρτηση **castVote**, η οποία δέχεται ως παράμετρο τη θέση της συγκεκριμένης ψηφοφορίας μέσα στον πίνακα polls και την θετική ή αρνητική ψήφο του χρήστη. Έπειτα, ελέγχει εάν η δοθείσα θέση είναι έγκυρη, εάν η ψηφοφορία είναι ανοικτή και αν ο χρήστης δεν έχει ήδη ψηφίσει. Εφόσον οι παραπάνω έλεγχοι ικανοποιούνται, αποθηκεύει την νέα ψήφο και αυξάνει κατά ένα την αντίστοιχη μεταβλητή που διατηρεί το πλήθος των θετικών/αρνητικών ψήφων.

```
/**
 * @notice Allows a user to cast their vote in a specific poll.
 * @dev Records the user's vote for the specified poll.
 * @param poll_idx The index of the poll in which the user is voting.
 * @param vote The vote (true/false) the user is casting.
 */
function castVote(uint poll_idx, bool vote) external {
    require(poll_idx < polls.length, "Invalid poll index"); // Check if poll_idx is within valid range

    Poll storage poll = polls[poll_idx]; // Get a reference to the selected poll
    require(poll.isOpen, "Poll is closed"); // Check if the poll is open for voting

    address userAddress = msg.sender;
    require(!poll.votes[userAddress], "Already voted"); // Check if the user has not already voted
    poll.votes[userAddress] = vote; // Record the user's vote

    // Increment the vote counters based on the vote value
    if (vote) {
        poll.numOfPositiveVotes++;
    } else {
        poll.numOfNegativeVotes++;
    }
}
```

Εικόνα 3: Συνάρτηση castVote

1.4 Συνάρτηση closePoll

Για το κλείσιμο μιας ψηφοφορίας υλοποιήθηκε η συνάρτηση **closePoll**, η οποία δέχεται ως παράμετρο τη θέση της συγκεκριμένης ψηφοφορίας μέσα στον πίνακα `polls`. Εάν η δοθείσα θέση είναι έγκυρη, θέτει την συγκεκριμένη ψηφοφορία ως κλειστή για να μην δέχεται άλλους ψήφους.

```
/**
 * @notice Closes a specific poll, preventing any further votes from being cast.
 * @dev Sets the isOpen status of the specified poll to false, ensuring it cannot be voted on.
 * @param poll_idx The index of the poll to close.
 */
function closePoll(uint poll_idx) external {
    require(poll_idx < polls.length, "Invalid poll index"); // Check if poll_idx is within valid range
    Poll storage poll = polls[poll_idx]; // Get a reference to the selected poll
    poll.isOpen = false; // Set isOpen to false
}
```

Εικόνα 4: Συνάρτηση closePoll

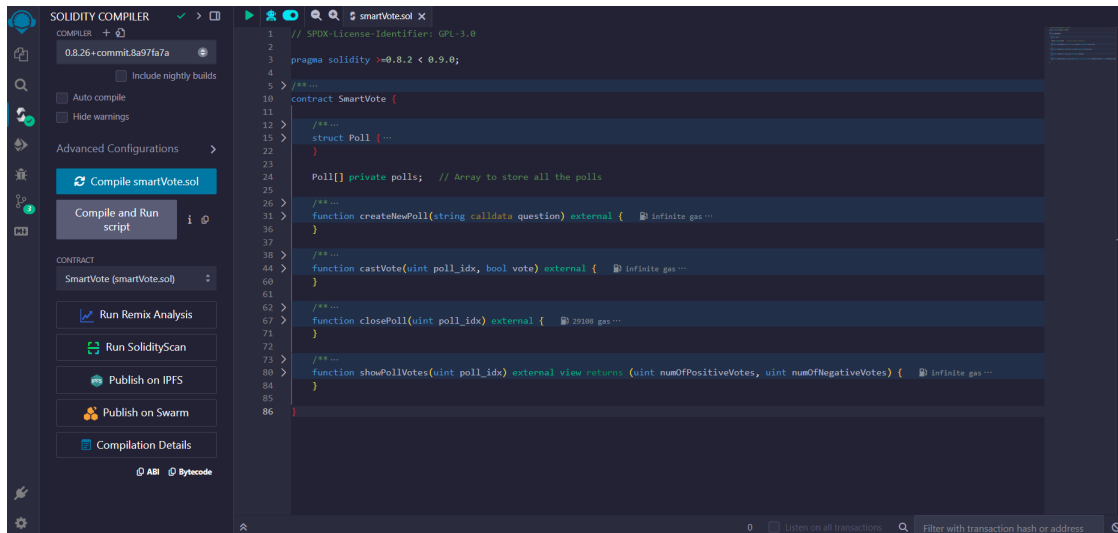
1.5 Συνάρτηση showPollVotes

Για την προβολή των αποτελεσμάτων μιας ψηφοφορίας υλοποιήθηκε η συνάρτηση **showPollVotes**, η οποία δέχεται ως παράμετρο τη θέση της συγκεκριμένης ψηφοφορίας μέσα στον πίνακα `polls`. Εάν η δοθείσα θέση είναι έγκυρη, επιστρέφεται το πλήθος θετικών και αρνητικών ψήφων της ψηφοφορίας.

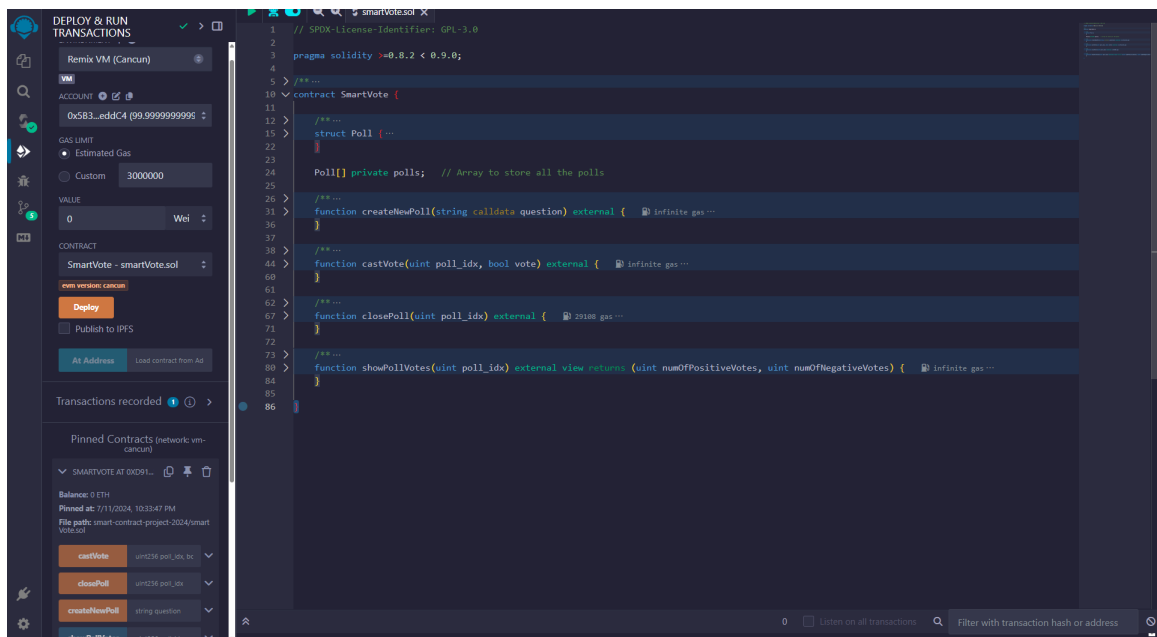
```
/**
 * @notice Returns the number of positive and negative votes for a specific poll.
 * @dev Retrieves the vote counts for the specified poll and returns them.
 * @param poll_idx The index of the poll whose votes are being queried.
 * @return numOfPositiveVotes The number of positive votes for the poll.
 * @return numOfNegativeVotes The number of negative votes for the poll.
 */
function showPollVotes(uint poll_idx) external view returns (uint numOfPositiveVotes, uint numOfNegativeVotes) {
    require(poll_idx < polls.length, "Invalid poll index"); // Check if poll_idx is within valid range
    Poll storage poll = polls[poll_idx]; // Get a reference to the selected poll
    return (poll.numOfPositiveVotes, poll.numOfNegativeVotes); // Return vote counts for positive and negative votes
}
```

Εικόνα 5: Συνάρτηση showPollVotes

2 Στιγμιότυπα Οθόνης

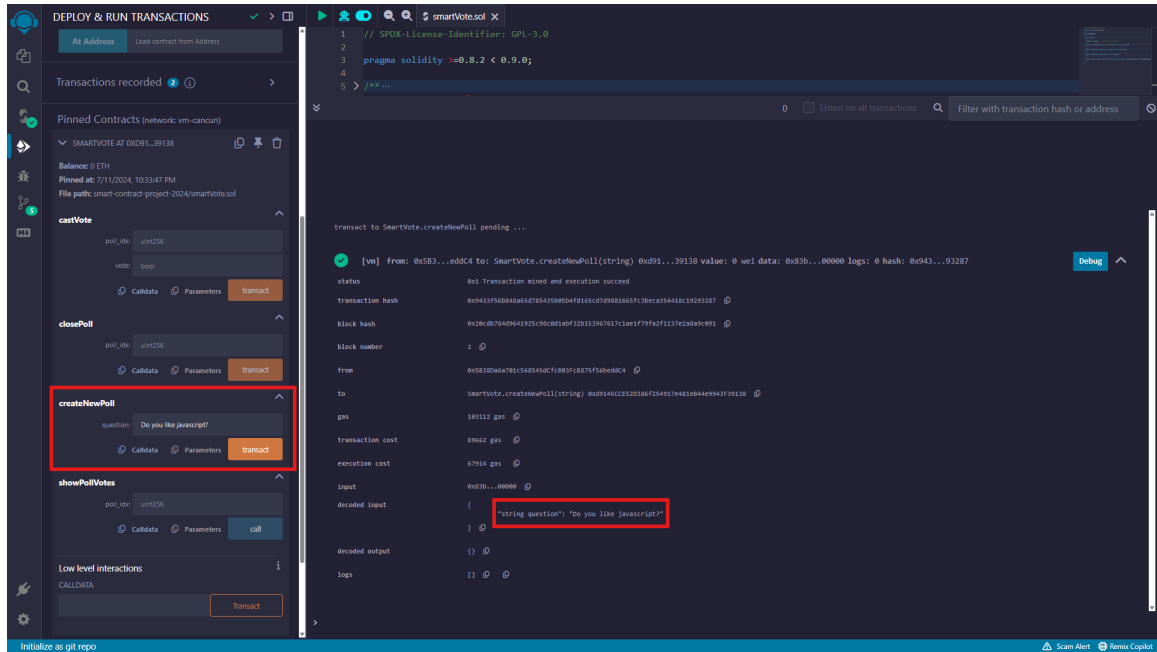


Εικόνα 6: Contract compilation

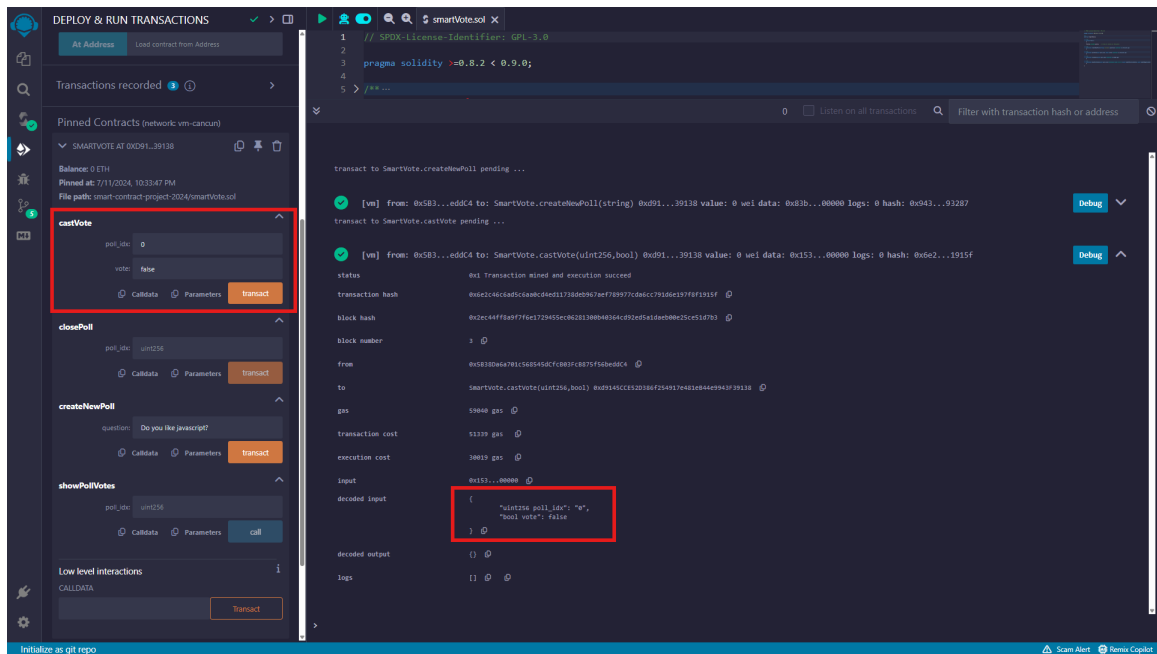


Εικόνα 7: Contract deployment

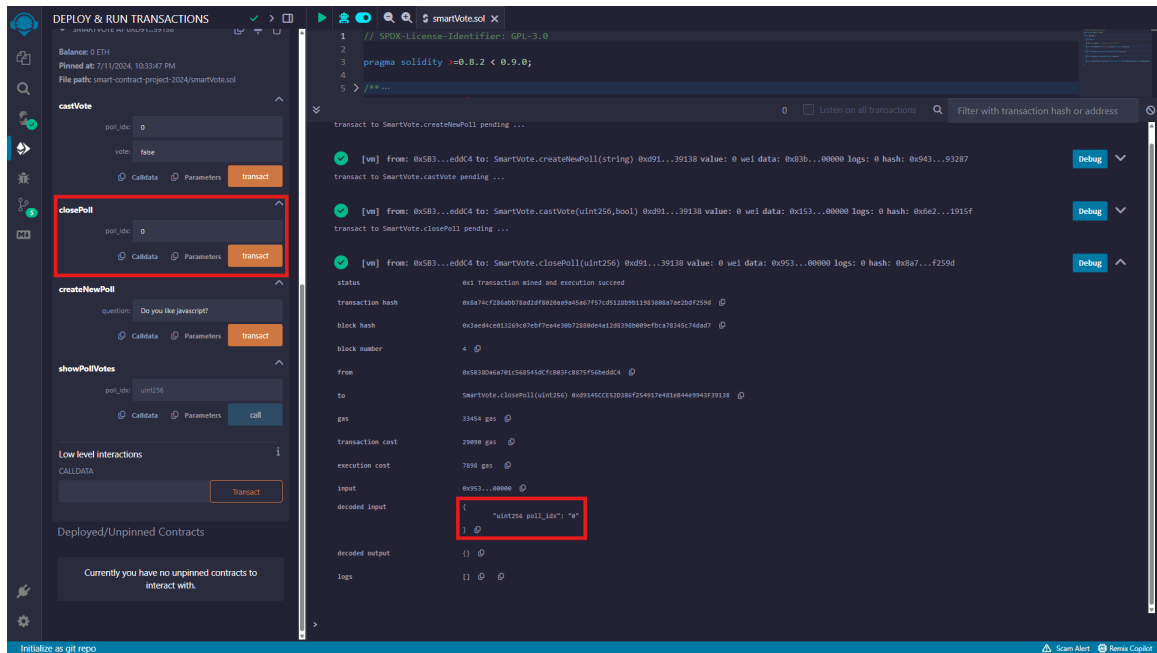
3 Παράδειγμα Εκτέλεσης



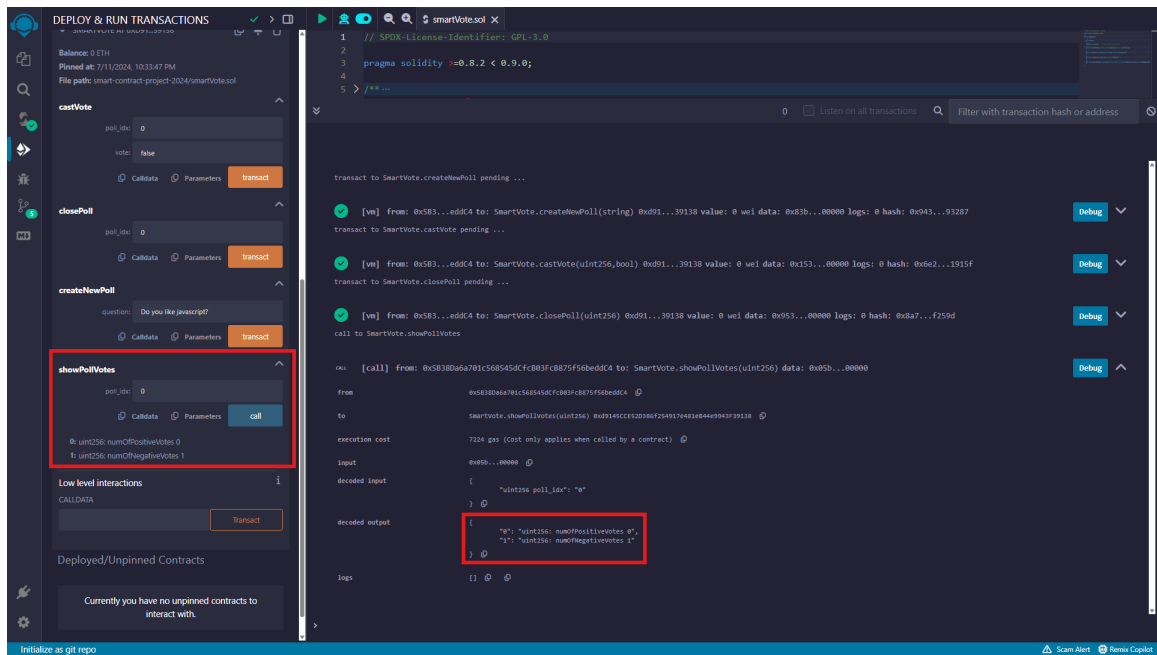
Εικόνα 8: Δημιουργία ψηφοφορίας



Εικόνα 9: Καταχώρηση ψήφου



Εικόνα 10: Κλείσιμο ψηφοφορίας



Εικόνα 11: Προβολή αποτελεσμάτων ψηφοφορίας