

```
<?php
require_once($src_root . '/lib/utenti.php');
require_once($src_root . '/lib/xml.php');

$doc_accrediti = load_xml('accrediti');

function crea_accredito($quantita) {
    global $doc_accrediti;

    $id_utente = $_SESSION['id_utente'];

    $root = $doc_accrediti->documentElement;
    $accredito = $root->childNodes;

    $id_accredito = get_next_id($accredito);

    $el_accredito = $doc_accrediti->createElement('accredito');
    $el_accredito->setAttribute('id', $id_accredito);

    $el_data = $doc_accrediti->createElement('data', date('Y-m-d\TH:i:s'));
    $el_accredito->appendChild($el_data);

    $el_id_utente = $doc_accrediti->createElement('idUtente', $id_utente);
    $el_accredito->appendChild($el_id_utente);

    $el_quantita = $doc_accrediti->createElement('quantita', $quantita);
    $el_accredito->appendChild($el_quantita);

    $root->appendChild($el_accredito);

    save_xml($doc_accrediti, 'accrediti');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_accrediti = load_xml('accrediti');

    return true;
}

function accetta_accredito($id_accredito) {
    global $doc_accrediti;

    $result = xpath($doc_accrediti, 'accrediti', '/ns:accrediti/ns:accredito[@id='
    . $id_accredito . ']');
    $accredito = $result[0];

    $ac_id_utente = $accredito->getElementsByTagName('idUtente')[0]->textContent;
    $ac_quantita = $accredito->getElementsByTagName('quantita')[0]->textContent;

    $accredito = $accredito->parentNode;
    $accredito->removeChild($accredito);

    save_xml($doc_accrediti, 'accrediti');

    global $doc_utenti;

    $result = xpath($doc_utenti, 'utenti', '/ns:utenti/ns:utente[@id=' . $ac_id_ute
    nte . ']');
    $utente = $result[0];

    $ut_credito = $utente->getElementsByTagName('credito')[0];
    $ut_credito->textContent += floatval($ac_quantita);
```

```
    save_xml($doc_utenti, 'utenti');

    return true;
}

function rifiuta_accredito($id_accredito) {
    global $doc_accrediti;

    $result = xpath($doc_accrediti, 'accrediti', '/ns:accrediti/ns:accredito[@id='
. $id_accredito . ']');
    $accredito = $result[0];

    $accrediti = $accredito->parentNode;
    $accrediti->removeChild($accredito);

    save_xml($doc_accrediti, 'accrediti');

    return true;
}
?>
```

```
<?php
require_once($src_root . '/lib/utenti.php');
require_once($src_root . '/lib/xml.php');

function aggiungi_carrello($id_prodotto, $qta_diff) {
    global $doc_utenti;

    $id_utente = $_SESSION['id_utente'];

    $result = xpath($doc_utenti, 'utenti',
        '/ns:utenti/ns:utente[@id=' . $id_utente . ']/ns:carrello/ns:prodotto[@id= '
        . $id_prodotto . ']'
    );

    if ($result->length === 0) {
        $result = xpath($doc_utenti, 'utenti',
            '/ns:utenti/ns:utente[@id=' . $id_utente . ']/ns:carrello'
        );
        $carrello = $result[0];

        $prodotto = $doc_utenti->createElement('prodotto');
        $prodotto->setAttribute('id', $id_prodotto);
        $prodotto->setAttribute('quantita', $qta_diff);
        $carrello->appendChild($prodotto);
    } else {
        $prodotto = $result[0];
        $qta_carrello = $prodotto->getAttribute('quantita');
        $qta_carrello += $qta_diff;
        $prodotto->setAttribute('quantita', $qta_carrello);
    }

    save_xml($doc_utenti, 'utenti');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_utenti = load_xml('utenti');

    return true;
}

function conta_carrello() {
    global $doc_utenti;

    if (!isset($_SESSION['id_utente'])) {
        return 0;
    }

    $id_utente = $_SESSION['id_utente'];

    $result = xpath($doc_utenti, 'utenti',
        '/ns:utenti/ns:utente[@id=' . $id_utente . ']/ns:carrello/ns:prodotto'
    );

    if ($result->length === 0) {
        return 0;
    }

    $somma = 0;
    foreach ($result as $item) {
        $somma += $item->getAttribute('quantita');
    }

    return $somma;
}
```

```
function modifica_carrello($id_prodotto, $quantita) {
    global $doc_utenti;

    $id_utente = $_SESSION['id_utente'];

    $result = xpath($doc_utenti, 'utenti',
        "/ns:utenti/ns:utente[@id='$id_utente']/ns:carrello/ns:prodotto[@id='$id_prodotto']"
    );

    $prodotto = $result[0];
    if ($quantita == 0) {
        $carrello = $prodotto->parentNode;
        $carrello->removeChild($prodotto);
    } else {
        $prodotto->setAttribute('quantita', $quantita);
    }

    save_xml($doc_utenti, 'utenti');

    return true;
}

function svuota_carrello() {
    global $doc_utenti;

    $id_utente = $_SESSION['id_utente'];

    $result = xpath($doc_utenti, 'utenti',
        "/ns:utenti/ns:utente[@id='$id_utente']/ns:carrello"
    );

    $carrello = $result[0];
    $utente = $carrello->parentNode;
    $utente->removeChild($carrello);

    $carrello = $doc_utenti->createElement('carrello');
    $utente->appendChild($carrello);

    save_xml($doc_utenti, 'utenti');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_utenti = load_xml('utenti');

    return true;
}
?>
```

```
<?php
require_once($src_root . '/lib/xml.php');

$doc_categorie = load_xml('categorie');

function ottieni_categoria($id_categoria) {
    global $doc_categorie;

    $result = xpath($doc_categorie, 'categorie', "/ns:categorie/ns:categoria[@id='$id_categoria']/ns:nome");
    if ($result->length != 1) {
        return "&lt;categoria sconosciuta&gt;";
    } else {
        return $result[0]->textContent;
    }
}

function elimina_categoria($id) {
    global $doc_categorie;

    $result = xpath($doc_categorie, 'categorie', '/ns:categorie/ns:categoria[@id='
. $id . ']');
    $categoria = $result[0];

    $categorie = $categoria->parentNode;
    $categoria->removeChild($categoria);

    save_xml($doc_categorie, 'categorie');

    return true;
}

function aggiungi_categoria($nome) {
    global $doc_categorie;

    $root = $doc_categorie->documentElement;
    $categorie = $root->childNodes;

    $id = get_next_id($categorie);

    $categoria = $doc_categorie->createElement('categoria');
    $categoria->setAttribute('id', $id);

    $nome = $doc_categorie->createElement('nome', $nome);
    $categoria->appendChild($nome);

    $root->appendChild($categoria);

    save_xml($doc_categorie, 'categorie');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_categorie = load_xml('categorie');

    return true;
}

function ordina_per_categoria($elements, $desc) {
    usort($elements, function($aElement, $bElement) use ($desc) {
        $aValue = $aElement->getElementsByTagName('categoria')[0]->textContent;
        $aValue = ottieni_categoria($aValue);

        $bValue = $bElement->getElementsByTagName('categoria')[0]->textContent;
        $bValue = ottieni_categoria($bValue);
```

```
    if ($desc) {  
        return strnatcmp($bValue, $aValue);  
    } else {  
        return strnatcmp($aValue, $bValue);  
    }  
});  
  
return $elements;  
}  
?>
```

```
<?php
const REGEX_CF = '/^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$/' ;
const REGEX_EMAIL = '/^[A-Za-z0-9._\-]+@[A-Za-zA0-9._\-]+\.[A-Za-z]+$/' ;
const REGEX_INDIRIZZO = '/^([[:alnum:]]+), ([a-zA-Z ]+), ([a-zA-Z ]+)$/' ;
const REGEX_PASSWORD = '/^[A-Za-z0-9!@£$%&()=?^,.;:_]{8,}$/' ;
const REGEX_TELEFONO = '/^\+[0-9]{12,13}$/' ;
?>
```

```
<?php
require_once($src_root . '/lib/rating.php');
require_once($src_root . '/lib/xml.php');

$doc_domande = load_xml('domande');

function elimina_domanda($id) {
    global $doc_domande;

    $result = xpath($doc_domande, 'domande', '/ns:domande/ns:domanda[@id=' . $id .
    ']'');
    $domanda = $result[0];

    $domande = $domanda->parentNode;
    $domande->removeChild($domanda);

    save_xml($doc_domande, 'domande');

    return true;
}

function elimina_domande($id_prodotto) {
    global $doc_domande;

    $domande = xpath($doc_domande, 'domande', '/ns:domande/ns:domanda[@idProdotto='
    . $id_prodotto . ']'');
    $root = $doc_domande->documentElement;

    foreach ($domande as $domanda) {
        $root->removeChild($domanda);
    }

    save_xml($doc_domande, 'domande');

    return true;
}

function aggiungi_domanda($id_prodotto, $contenuto_dom) {
    global $doc_domande;
    $id_ut = $_SESSION['id_utente'];

    $root = $doc_domande->documentElement;
    $domande = $root->childNodes;

    $nuova_domanda = $doc_domande->createElement('domanda');

    $id = get_next_id($domande);
    $nuova_domanda->setAttribute('id', $id);
    $nuova_domanda->setAttribute('idProdotto', $id_prodotto);

    $sel_u = $doc_domande->createElement('idUtente', $id_ut);
    $nuova_domanda->appendChild($sel_u);

    $contenuto = $doc_domande->createElement('contenuto', $contenuto_dom);
    $nuova_domanda->appendChild($contenuto);

    $data = $doc_domande->createElement('data', date('Y-m-d'));
    $nuova_domanda->appendChild($data);

    $sel_ratings = $doc_domande->createElement('ratings');
    $nuova_domanda->appendChild($sel_ratings);

    $sel_risposte = $doc_domande->createElement('risposte');
    $nuova_domanda->appendChild($sel_risposte);
}
```



```
$root->appendChild($nuova_domanda);

save_xml($doc_domande, 'domande');

// BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
documento vecchio
$doc_domande = load_xml('domande');

return true;
}

function aggiungi_rating_domanda($id_domanda, $supporto, $utilita) {
    global $doc_domande;

    $result = xpath($doc_domande, 'domande', "/ns:domande/ns:domanda[@id='$id_domanda']/ns:ratings");
    $ratings = $result[0];

    aggiungi_rating($doc_domande, $ratings, $supporto, $utilita);

    save_xml($doc_domande, 'domande');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_domande = load_xml('domande');

    return true;
}

function elimina_risposta($id_domanda, $id_risposta) {
    global $doc_domande;

    $result = xpath($doc_domande, 'domande', '/ns:domande/ns:domanda[@id=' . $id_domanda . ']/ns:risposte/ns:risposta[@id=' . $id_risposta . ']');
    $risposta = $result[0];

    $risposte = $risposta->parentNode;
    $risposte->removeChild($risposta);

    save_xml($doc_domande, 'domande');

    return true;
}

function aggiungi_risposta($id_domanda, $contenuto_r) {
    global $doc_domande;

    $risposte = xpath($doc_domande, 'domande', '/ns:domande/ns:domanda[@id=' . $id_domanda . ']/ns:risposte')[0];

    $nuova_risposta = $doc_domande->createElement('risposta');

    $arr_risposte = $risposte->childNodes;
    $id = get_next_id($arr_risposte);
    $nuova_risposta->setAttribute('id', $id);

    $id_ut = $_SESSION['id_utente'];
    $sel_id_ut = $doc_domande->createElement('idUtente', $id_ut);
    $nuova_risposta->appendChild($sel_id_ut);

    $sel_contenuto = $doc_domande->createElement('contenuto', $contenuto_r);
    $nuova_risposta->appendChild($sel_contenuto);
```

```
$el_data = $doc_domande->createElement('data', date('Y-m-d'));
$nuova_risposta->appendChild($el_data);

$el_ratings = $doc_domande->createElement('ratings');
$nuova_risposta->appendChild($el_ratings);

$risposte->appendChild($nuova_risposta);

save_xml($doc_domande, 'domande');

// BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
documento vecchio
$doc_domande = load_xml('domande');

return true;
}

function aggiungi_rating_risposta($id_domanda, $id_risposta, $supporto, $utilita)
{
    global $doc_domande;

    $result = xpath($doc_domande, 'domande', "/ns:domande/ns:domanda[@id='$id_domanda']/ns:risposte/ns:risposta[@id='$id_risposta']/ns:ratings");
    $ratings = $result[0];

    aggiungi_rating($doc_domande, $ratings, $supporto, $utilita);

    save_xml($doc_domande, 'domande');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_domande = load_xml('domande');

    return true;
}

function presenza_gestore_risposta($id_domanda) {
    global $doc_domande;
    $trovato = false;

    $domande = xpath($doc_domande, 'domande', '/ns:domande/ns:domanda[@id=' . $id_domanda . ']');

    foreach ($domande as $domanda) {
        $risposte = $domanda->getElementsByTagName('risposte')[0]->childNodes;
        foreach ($risposte as $risposta) {
            $id_ut_r = $risposta->getElementsByTagName('idUtente')[0]->textContent;
            $trovato = trova_gestore($id_ut_r);
            if ($trovato) {
                return true;
            }
        }
    }

    return false;
}
?>
```

```
<?php
require_once($src_root . '/lib/xml.php');

$doc_faq = load_xml('faq');

function aggiungi_faq($domanda, $risposta) {
    global $doc_faq;

    $root = $doc_faq->documentElement;
    $faqs = $root->childNodes;

    $id = get_next_id($faqs);

    $faq = $doc_faq->createElement('faq');
    $faq->setAttribute('id', $id);

    $domanda = $doc_faq->createElement('domanda', $domanda);
    $faq->appendChild($domanda);

    $risposta = $doc_faq->createElement('risposta', $risposta);
    $faq->appendChild($risposta);

    $root->appendChild($faq);

    save_xml($doc_faq, 'faq');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_faq = load_xml('faq');

    return true;
}

function elimina_faq($id) {
    global $doc_faq;

    $result = xpath($doc_faq, 'faq', '/ns:faqs/ns:faq[@id=' . $id . ']');
    $faq = $result[0];

    $faqs = $faq->parentNode;
    $faqs->removeChild($faq);

    save_xml($doc_faq, 'faq');

    return true;
}

function modifica_faq($id, $domanda, $risposta) {
    global $doc_faq;

    $result = xpath($doc_faq, 'faq', '/ns:faqs/ns:faq[@id=' . $id . ']');
    $faq = $result[0];

    $faq->getElementsByTagName('domanda')[0]->textContent = $domanda;
    $faq->getElementsByTagName('risposta')[0]->textContent = $risposta;

    save_xml($doc_faq, 'faq');

    return true;
}
?>
```

```
<hr id="riga-sezioni" />
<div id="footer" class="centrato">
  <div id="sopra" class="py-1em">
    <div>
      <h3 class="pb-8">Contatti</h3>
      <p id="contatti">
        &#x01F4E7 cecchetto.1941039@studenti.uniroma1.it<br>
        &#x01F4E7 roccia.1967318@studenti.uniroma1.it<br>
        &#x01F4DE +39 07731234567
      </p>
    </div>
    <div>
      <h3 class="pb-8">Dove siamo</h3>
      &#x01F4CC Via XXIV Maggio, 7, 04100 Latina LT
    </div>
  </div>
  <div id="sotto" class="py-1em">
    Copyright R&C store 2023
  </div>
</div>
```

```
<?php
require_once($src_root . '/lib/carrello.php');

$qta_carrello = conta_carrello();
?>
<div id="header">
    <div id="icone">
        <span>
            <a href="<?php echo($src_subdir); ?>/utente/index.php" title="Dashboard">&#x
01F464</a>
        </span>
    <?php if ($e_visitatore || $e_cliente) { ?>
        <span class="icona-con-notifica">
            <a href="<?php echo($src_subdir); ?>/cliente/carrello.php" title="Carrello">
&#x01F6D2</a>
            <span><?php echo($qta_carrello); ?></span>
        </span>
    <?php } ?>
    </div>
    <h1 id="titolo">
        <a href="<?php echo($src_subdir); ?>/index.php">
            
        </a>
    </h1>
    <ul id="menu">
        <li><a href="<?php echo($src_subdir); ?>/index.php">Homepage</a></li>
        <li><a href="<?php echo($src_subdir); ?>/catalogo.php">Catalogo</a></li>
        <li><a href="<?php echo($src_subdir); ?>/faq.php">FAQ</a></li>
        <li><a href="<?php echo($src_subdir); ?>/info.php">Informazioni</a></li>
    </ul>
</div>
<hr id="riga-sezioni" />
```

```
<?php
require_once($src_root . '/lib/utenti.php');
require_once($src_root . '/lib/xml.php');

$doc_offerte = load_xml('offerte');

function calcola_sconto($offerte) {
    $sconto_tot = 0.00;

    foreach ($offerte as $offerta) {
        $tipo = $offerta->getElementsByTagName('tipo')[0]->textContent;
        if ($tipo !== 'sconto') {
            continue;
        }

        $sconto_off = $offerta->getElementsByTagName('percentuale')[0]->textContent;
        $sconto_tot += $sconto_off / 100;
    }

    return $sconto_tot;
}

function calcola_bonus($offerte) {
    $bonus_tot = 0;

    foreach ($offerte as $offerta) {
        $tipo = $offerta->getElementsByTagName('tipo')[0]->textContent;
        if ($tipo !== 'bonus') {
            continue;
        }

        $bonus_off = $offerta->getElementsByTagName('numCrediti')[0]->textContent;
        $bonus_tot += $bonus_off;
    }

    return $bonus_tot;
}

function offerte_applicabili($prodotto) {
    global $doc_offerte;
    global $doc_utenti;

    $id_utente = isset($_SESSION['id_utente']) ? $_SESSION['id_utente'] : 0;

    $root = $doc_offerte->documentElement;
    $offerte = $root->childNodes;

    $off_app = [];

    foreach ($offerte as $offerta) {
        $data_inizio = '1970-01-01';
        $target = $offerta->getElementsByTagName('target')[0]->textContent;

        switch ($target) {
            case 'credData':
                // "clienti che hanno speso X crediti da una certa data"
                $data_inizio = $offerta->getElementsByTagName('dataInizio')[0]->textContent;
            case 'credInizio':
                if ($id_utente === 0) {
                    break;
                }
                $cred_spesi_off = $offerta->getElementsByTagName('creditiSpesi')[0]->textContent;
        }
    }
}
```

```
$cred_spesi_att = calcola_crediti_spesi($id_utente, $data_inizio);
if ($cred_spesi_att >= $cred_spesi_off) {
    array_push($off_app, $offerta);
}
break;
case 'reputazione':
    // "clienti che hanno una reputazione >= X"
    if ($id_utente === 0) {
        break;
    }
    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id='$id_utente']/ns:reputazione");
    if ($result->length === 0) {
        break;
    }
    $result = $result[0]->textContent;
    $reputazione = $offerta->getElementsByTagName('reputazione')[0]->textContent;

    if ($result >= $reputazione) {
        array_push($off_app, $offerta);
    }
    break;
case 'dataReg':
    // "clienti che sono con noi da X anni"
    if ($id_utente === 0) {
        break;
    }
    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id='$id_utente']/ns:dataRegistrazione");
    if ($result->length === 0) {
        break;
    }
    $result = $result[0]->textContent;

    $data_reg = date_create($result);
    $oggi = date_create();
    $diff = date_diff($data_reg, $oggi)->format('%Y');
    $anni = $offerta->getElementsByTagName('anni')[0]->textContent;

    if ($diff >= $anni) {
        array_push($off_app, $offerta);
    }
    break;
case 'prodSpec':
    // "e' un prodotto particolare"
    $id_prod_prod = $prodotto->getAttribute('id');
    $id_prod_off = $offerta->getElementsByTagName('idProdotto')[0]->textContent;

    if ($id_prod_off == $id_prod_prod) {
        array_push($off_app, $offerta);
    }
    break;
case 'categoria':
    // "il prodotto e' di una determinata categoria"
    $id_cat_prod = $prodotto->getElementsByTagName('categoria')[0]->textContent;
    $id_cat_off = $offerta->getElementsByTagName('idCategoria')[0]->textContent;

    if ($id_cat_off == $id_cat_prod) {
        array_push($off_app, $offerta);
    }
    break;
case 'eccMag':
    // "e' presente in magazzino un'eccezione del prodotto"
```

```
$id_prod_prod = $prodotto->getAttribute('id');
$qta_prod = $prodotto->getElementsByTagName('quantita')[0]->textContent;

$id_prod_off = $offerta->getElementsByTagName('idProdotto')[0]->textConte
nt;
$qta_off = $offerta->getElementsByTagName('quantitaMin')[0]->textContent;

if ($id_prod_off == $id_prod_prod && $qta_prod >= $qta_off) {
    array_push($off_app, $offerta);
}
break;
}
}

return $off_app;
}

function aggiungi_offerta($id, $campi) {
    global $doc_offerte;

    $root = $doc_offerte->documentElement;

    $offerte = $root->childNodes;
    if ($id == 0) {
        $id = get_next_id($offerte);
    }

    $offerta = $doc_offerte->createElement('offerta');
    $offerta->setAttribute('id', $id);

    $tipo = $doc_offerte->createElement('tipo', $campi['tipo']);
    $offerta->appendChild($tipo);

    if ($campi['tipo'] === 'sconto') {
        $percentuale = $doc_offerte->createElement('percentuale', $campi['percentuale
    ']);
        $offerta->appendChild($percentuale);
    } else if ($campi['tipo'] === 'bonus') {
        $numCrediti = $doc_offerte->createElement('numCrediti', $campi['numCrediti'])
    ;
        $offerta->appendChild($numCrediti);
    }

    $target = $doc_offerte->createElement('target', $campi['target']);
    $offerta->appendChild($target);

    switch ($campi['target']) {
        case 'credData':
            $creditiSpesi = $doc_offerte->createElement('creditiSpesi', $campi['crediti
            Spesi']);
            $offerta->appendChild($creditiSpesi);
            $dataInizio = $doc_offerte->createElement('dataInizio', $campi['dataInizio
        ']);
            $offerta->appendChild($dataInizio);
            break;

        case 'reputazione':
            $reputazione = $doc_offerte->createElement('reputazione', $campi['reputazio
            ne']);
            $offerta->appendChild($reputazione);
            break;
    }
}
```



```
    case 'dataReg':
        $anni = $doc_offerte->createElement('anni', $campi['anni']);
        $offerta->appendChild($anni);
        break;

    case 'prodSpec':
        $idProdotto = $doc_offerte->createElement('idProdotto', $campi['idProdotto'
]);
        $offerta->appendChild($idProdotto);
        break;

    case 'categoria':
        $idCategoria = $doc_offerte->createElement('idCategoria', $campi['idCategor
ia']);
        $offerta->appendChild($idCategoria);
        break;

    case 'eccMag':
        $idProdotto = $doc_offerte->createElement('idProdotto', $campi['idProdotto'
]);
        $offerta->appendChild($idProdotto);
        $quantitaMin = $doc_offerte->createElement('quantitaMin', $campi['quantitaM
in']);
        $offerta->appendChild($quantitaMin);
        break;
    }

    $root->appendChild($offerta);

    save_xml($doc_offerte, 'offerte');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
documento vecchio
    $doc_offerte = load_xml('offerte');

    return true;
}

function modifica_offerta($id, $campi) {
    elimina_offerta($id);
    aggiungi_offerta($id, $campi);
}

function elimina_offerta($id) {
    global $doc_offerte;

    $result = xpath($doc_offerte, 'offerte', '/ns:offerte/ns:offerta[@id=' . $id .
']');
    $offerta = $result[0];

    $offerta = $offerta->parentNode;
    $offerta->removeChild($offerta);

    save_xml($doc_offerte, 'offerte');

    return true;
}

function elimina_offerte_prodotto($id_prodotto) {
    global $doc_offerte;

    $root = $doc_offerte->documentElement;
    $offerta = $root->childNodes;
```

```
$da_eliminare = [];  
  
foreach ($offerte as $offerta) {  
    $target = $offerta->getElementsByTagName('target')[0]->textContent;  
  
    if ($target !== 'prodSpec' && $target !== 'eccMag') {  
        continue;  
    }  
  
    $id_prod_off = $offerta->getElementsByTagName('idProdotto')[0]->textContent;  
  
    if ($id_prod_off == $id_prodotto) {  
        array_push($da_eliminare, $offerta);  
    }  
}  
  
foreach ($da_eliminare as $offerta) {  
    $root->removeChild($offerta);  
}  
  
save_xml($doc_offerte, 'offerte');  
  
return true;  
}  
  
function ottieni_info_offerta($id) {  
    global $doc_offerte;  
  
    $result = xpath($doc_offerte, 'offerte', "/ns:offerte/ns:offerta[@id=$id]");  
    if ($result->length !== 1) {  
        return null;  
    }  
  
    $offerta = $result[0];  
  
    $result = [  
        'percentuale' => 0,  
        'numCrediti' => 0,  
        'creditiSpesi' => 0,  
        'dataInizio' => '1970-01-01',  
        'reputazione' => 0,  
        'anni' => 0,  
        'idProdotto' => 1,  
        'idCategoria' => 1,  
        'quantitaMin' => 0,  
    ];  
  
    $result['tipo'] = $offerta->getElementsByTagName('tipo')[0]->textContent;  
  
    if ($result['tipo'] === 'sconto') {  
        $result['percentuale'] = $offerta->getElementsByTagName('percentuale')[0]->te  
xtContent;  
    } else if ($result['tipo'] === 'bonus') {  
        $result['numCrediti'] = $offerta->getElementsByTagName('numCrediti')[0]->text  
Content;  
    }  
  
    $result['target'] = $offerta->getElementsByTagName('target')[0]->textContent;  
  
    switch ($result['target']) {  
        case 'credData':  
            $result['creditiSpesi'] = $offerta->getElementsByTagName('creditiSpesi')[0]  
->textContent;  
            $result['dataInizio'] = $offerta->getElementsByTagName('dataInizio')[0]->te
```

```
xtContent;
    break;

    case 'reputazione':
        $result['reputazione'] = $offerta->getElementsByTagName('reputazione')[0]->
textContent;
        break;

    case 'dataReg':
        $result['anni'] = $offerta->getElementsByTagName('anni')[0]->textContent;
        break;

    case 'prodSpec':
        $result['idProdotto'] = $offerta->getElementsByTagName('idProdotto')[0]->te
xtContent;
        break;

    case 'categoria':
        $result['idCategoria'] = $offerta->getElementsByTagName('idCategoria')[0]->
textContent;
        break;

    case 'eccMag':
        $result['idProdotto'] = $offerta->getElementsByTagName('idProdotto')[0]->te
xtContent;
        $result['quantitaMin'] = $offerta->getElementsByTagName('quantitaMin')[0]->
textContent;
        break;
    }

    return $result;
}
?>
```

```
<?php
require_once($src_root . '/lib/xml.php');

$doc_ordini = load_xml('ordini');

function crea_ordine($indirizzo, $prezzo, $prodotti) {
    global $doc_ordini;

    $id_utente = $_SESSION['id_utente'];

    $root = $doc_ordini->documentElement;
    $ordini = $root->childNodes;

    $id_ordine = get_next_id($ordini);

    $nuovo_ordine = $doc_ordini->createElement('ordine');

    $nuovo_ordine->setAttribute('id', $id_ordine);
    $nuovo_ordine->setAttribute('idUtente', $id_utente);

    $el_data = $doc_ordini->createElement('data', date('Y-m-d H:i:s'));
    $nuovo_ordine->appendChild($el_data);

    $el_indirizzo = $doc_ordini->createElement('indirizzo', $indirizzo);
    $nuovo_ordine->appendChild($el_indirizzo);

    $el_prezzo = $doc_ordini->createElement('prezzoFinale', $prezzo);
    $nuovo_ordine->appendChild($el_prezzo);

    $el_prodotti = $doc_ordini->createElement('prodotti');

    foreach ($prodotti as $prodotto) {
        $nuovo_prodotto = $doc_ordini->createElement('prodotto');

        $id_prod = $prodotto->getAttribute('id');
        $qta_prod = $prodotto->getAttribute('quantita');

        $nuovo_prodotto->setAttribute('id', $id_prod);
        $nuovo_prodotto->setAttribute('quantita', $qta_prod);

        $el_prodotti->appendChild($nuovo_prodotto);
    }

    $nuovo_ordine->appendChild($el_prodotti);

    $root->appendChild($nuovo_ordine);

    save_xml($doc_ordini, 'ordini');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_ordini = load_xml('ordini');

    return true;
}

function calcola_crediti_spesi($id_utente, $data_inizio) {
    global $doc_ordini;

    $ordini = xpath($doc_ordini, 'ordini', "/ns:ordini/ns:ordine[@idUtente=$id_uten
te]");

    $cred_spesi = 0;
    $data_inizio = date_create($data_inizio);
```

```
foreach ($ordini as $ordine) {
    $data_ord = $ordine->getElementsByTagName('data')[0]->textContent;
    $data_ord = date_create($data_ord);
    if ($data_ord < $data_inizio) {
        continue;
    }

    $prezzo_finale = $ordine->getElementsByTagName('prezzoFinale')[0]->textContent;
    $cred_spesi += $prezzo_finale;
}

return $cred_spesi;
}
?>
```

```
<?php
require_once($src_root . '/lib/domande.php');
require_once($src_root . '/lib/offerte.php');
require_once($src_root . '/lib/recensioni.php');
require_once($src_root . '/lib/xml.php');

$doc_prodotti = load_xml('prodotti');

function aggiungi_prodotto($nome, $marca, $descrizione, $costo, $categoria, $quantita, $immagine) {
    global $doc_prodotti;

    $root = $doc_prodotti->documentElement;
    $prodotti = $root->childNodes;

    $nuovo_prodotto = $doc_prodotti->createElement('prodotto');

    $id = get_next_id($prodotti);
    $nuovo_prodotto->setAttribute('id', $id);

    $el_marca = $doc_prodotti->createElement('marca', $marca);
    $nuovo_prodotto->appendChild($el_marca);

    $el_nome = $doc_prodotti->createElement('nome', $nome);
    $nuovo_prodotto->appendChild($el_nome);

    $el_descrizione = $doc_prodotti->createElement('descrizione', $descrizione);
    $nuovo_prodotto->appendChild($el_descrizione);

    $el_costo = $doc_prodotti->createElement('costo', $costo);
    $nuovo_prodotto->appendChild($el_costo);

    $el_categoria = $doc_prodotti->createElement('categoria', $categoria);
    $nuovo_prodotto->appendChild($el_categoria);

    $el_quantita = $doc_prodotti->createElement('quantita', $quantita);
    $nuovo_prodotto->appendChild($el_quantita);

    $root->appendChild($nuovo_prodotto);

    save_xml($doc_prodotti, 'prodotti');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_prodotti = load_xml('prodotti');

    $upload_ok = isset($immagine['error']) &&
        $immagine['error'] === UPLOAD_ERR_OK;
    if ($upload_ok) {
        $nome_temp = $immagine['tmp_name'];
        $nome_finale = $src_root . '/res/img/prodotti/' . $id . '.png';
        move_uploaded_file($nome_temp, $nome_finale);
    }

    return true;
}

function elimina_prodotto($id) {
    global $doc_prodotti;

    $result = xpath($doc_prodotti, 'prodotti', '/ns:prodotti/ns:prodotto[@id=' . $id . ']');
    $prodotto = $result[0];
```

```
$prodotti = $prodotto->parentNode;
$prodotti->removeChild($prodotto);

save_xml($doc_prodotti, 'prodotti');

elimina_recensioni($id);
elimina_domande($id);
elimina_offerte_prodotto($id);
}

function modifica_prodotto($id, $nome, $marca, $descrizione, $costo, $categoria,
$quantita, $immagine) {
    global $doc_prodotti;

    $result = xpath($doc_prodotti, 'prodotti', "/ns:prodotti/ns:prodotto[@id=$id]")
;
    $prodotto = $result[0];

    $prodotto->getElementsByTagName('nome')[0]->textContent = $nome;
    $prodotto->getElementsByTagName('marca')[0]->textContent = $marca;
    $prodotto->getElementsByTagName('descrizione')[0]->textContent = $descrizione;
    $prodotto->getElementsByTagName('costo')[0]->textContent = $costo;
    $prodotto->getElementsByTagName('categoria')[0]->textContent = $categoria;
    $prodotto->getElementsByTagName('quantita')[0]->textContent = $quantita;

    save_xml($doc_prodotti, 'prodotti');

    $upload_ok = isset($immagine['error']) &&
        $immagine['error'] === UPLOAD_ERR_OK;
    if ($upload_ok) {
        $nome_temp = $immagine['tmp_name'];
        $nome_finale = $src_root . '/res/img/prodotti/' . $p_id . '.png';
        move_uploaded_file($nome_temp, $nome_finale);
    }

    return true;
}

function ottieni_info_prodotto($id) {
    global $doc_prodotti;

    $result = xpath($doc_prodotti, 'prodotti', "/ns:prodotti/ns:prodotto[@id=$id]")
;
    if ($result->length !== 1) {
        return null;
    }

    $prodotto = $result[0];

    $nome = $prodotto->getElementsByTagName('nome')[0]->textContent;
    $marca = $prodotto->getElementsByTagName('marca')[0]->textContent;
    $descrizione = $prodotto->getElementsByTagName('descrizione')[0]->textContent;
    $costo = $prodotto->getElementsByTagName('costo')[0]->textContent;
    $categoria = $prodotto->getElementsByTagName('categoria')[0]->textContent;
    $quantita = $prodotto->getElementsByTagName('quantita')[0]->textContent;

    return [
        'nome' => $nome,
        'marca' => $marca,
        'descrizione' => $descrizione,
        'costo' => $costo,
        'categoria' => $categoria,
        'quantita' => $quantita,
    ];
};
```

`}`

```
function scala_qta_prodotto($id, $qta_diff) {  
    global $doc_prodotti;
```

```
    $quantita = xpath($doc_prodotti, 'prodotti', '/ns:prodotti/ns:prodotto[@id=' .  
$id . ']/ns:quantita')[0];
```

```
    $quantita->textContent -= $qta_diff;
```

```
    save_xml($doc_prodotti, 'prodotti');
```

```
    return true;
```

`}``?>`


```
<?php
require_once($src_root . '/lib/ordini.php');
require_once($src_root . '/lib/utenti.php');

const K_CLIENTE = 1;
const K_ACQUIRENTE = 2;
const K_GESTORE = 3;

// m = 1000 / ln(4)
const SCALA_EXP = 721.35;

function calcola_rating_medio($ratings) {
    if ($ratings->length === 0) {
        return [
            'supporto' => 0.00,
            'utilita' => 0.00
        ];
    }

    $acc_supp = 0.00;
    $acc_util = 0.00;

    for ($i = 0; $i < $ratings->length; $i++) {
        $rating = $ratings[$i];

        $rat_supp = $rating->getElementsByTagName('supporto')[0]->textContent;
        $rat_util = $rating->getElementsByTagName('utilita')[0]->textContent;

        $acc_supp += $rat_supp;
        $acc_util += $rat_util;
    }

    return [
        'supporto' => $acc_supp / $ratings->length,
        'utilita' => $acc_util / $ratings->length
    ];
}

function aggiorna_reputazione($id_ut_dest, $id_prod, $supporto, $utilita) {
    global $doc_ordini;
    global $doc_utenti;

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id='$id_ut_dest']
/ns:reputazione");
    $rep_el = $result[0];
    $rep_dest = $rep_el->textContent;

    if ($rep_dest == 0) {
        return;
    }

    $k_funzione = K_CLIENTE;

    switch ($_SESSION['tipo_utente']) {
        case 'cliente':
            $id_ut_mitt = $_SESSION['id_utente'];

            $result = xpath($doc_ordini, 'ordini',
                "/ns:ordini/ns:ordine[@idUtente='$id_ut_mitt']/ns:prodotti/ns:prodotto[@i
d='$id_prod']");

            if ($result->length !== 0) {
```

```

    $k_funzione = K_ACQUIRENTE;
}

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id='$sid_ut_mitt']/ns:reputazione");
    $rep_mitt = $result[0]->textContent;

    $k_funzione = K_GESTORE - (K_GESTORE - $k_funzione) * exp(-$rep_mitt / SCALA_EXP);
    $k_funzione = round($k_funzione, 2);

    break;
case 'gestore':
    $k_funzione = K_GESTORE;
    break;
}

    $var_supporto = $k_funzione * (2 * $supporto - 4); // (1..3) => k * (-2, 0, +2)
    $var_utilita = $k_funzione * (2 * $utilita - 6); // (1..5) => k * (-4, -2, 0, +2, +4)

    $var_tot = $var_utilita + $var_supporto;
    $rep_dest += $var_tot;

    if ($rep_dest < 0) {
        $rep_el->textContent = 0;
    } else {
        $rep_el->textContent = $rep_dest;
    }

    save_xml($doc_utenti, 'utenti');

    return true;
}

function aggiungi_rating($doc, $ratings, $supporto, $utilita) {
    $id_utente = $_SESSION['id_utente'];

    $rating = $doc->createElement('rating');

    $rating->setAttribute('idUtente', $id_utente);

    $el_supporto = $doc->createElement('supporto', $supporto);
    $rating->appendChild($el_supporto);

    $el_utilita = $doc->createElement('utilita', $utilita);
    $rating->appendChild($el_utilita);

    $ratings->appendChild($rating);

    return true;
}
?>

```

```
<?php
require_once($src_root . '/lib/rating.php');
require_once($src_root . '/lib/xml.php');

$doc_recensioni = load_xml('recensioni');

function aggiungi_recensione($id_prodotto, $contenuto) {
    global $doc_recensioni;

    $id_utente = $_SESSION['id_utente'];

    $root = $doc_recensioni->documentElement;
    $recensioni = $root->childNodes;

    $nuova_recensione = $doc_recensioni->createElement('recensione');

    $id_recensione = get_next_id($recensioni);

    $nuova_recensione->setAttribute('id', $id_recensione);
    $nuova_recensione->setAttribute('idProdotto', $id_prodotto);

    $el_idUt = $doc_recensioni->createElement('idUtente', $id_utente);
    $nuova_recensione->appendChild($el_idUt);

    $el_contenuto = $doc_recensioni->createElement('contenuto', $contenuto);
    $nuova_recensione->appendChild($el_contenuto);

    $el_data = $doc_recensioni->createElement('data', date('Y-m-d'));
    $nuova_recensione->appendChild($el_data);

    $el_ratings = $doc_recensioni->createElement('ratings');
    $nuova_recensione->appendChild($el_ratings);

    $root->appendChild($nuova_recensione);

    save_xml($doc_recensioni, 'recensioni');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_recensioni = load_xml('recensioni');

    return true;
}

function aggiungi_rating_recensione($id_recensione, $supporto, $utilita) {
    global $doc_recensioni;

    $result = xpath($doc_recensioni, 'recensioni', "/ns:recensioni/ns:recensione[@i
d='$id_recensione']/ns:ratings");
    $ratings = $result[0];

    aggiungi_rating($doc_recensioni, $ratings, $supporto, $utilita);

    save_xml($doc_recensioni, 'recensioni');

    // BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
    documento vecchio
    $doc_recensioni = load_xml('recensioni');

    return true;
}

function elimina_recensione($id) {
    global $doc_recensioni;
```

```
$result = xpath($doc_recensioni, 'recensioni', '/ns:recensioni/ns:recensione[@id=' . $id . ']'');
$recensione = $result[0];

$recensioni = $recensione->parentNode;
$recensioni->removeChild($recensione);

save_xml($doc_recensioni, 'recensioni');

return true;
}

function elimina_recensioni($id_prodotto) {
    global $doc_recensioni;

    $recensioni = xpath($doc_recensioni, 'recensioni', '/ns:recensioni/ns:recensione[@idProdotto=' . $id_prodotto . ']'');
    $root = $doc_recensioni->documentElement;

    foreach ($recensioni as $recensione) {
        $root->removeChild($recensione);
    }

    save_xml($doc_recensioni, 'recensioni');

    return true;
}
?>
```

```
<?php
$src_root = realpath(__DIR__ . '/../..');

require_once($src_root . '/lib/utills.php');

$src_subdir = make_subdir($src_level);

session_start();

// Caso particolare
/*
 * La pagina del carrello riceve un post quando si aggiunge un elemento.
 * Serve memorizzare i dati del post anche quando l'utente non e' loggato per
 * permettere l'aggiunta dell'elemento dopo il login.
 * Quindi questa operazione va in questo file, perche' va eseguita dopo
 * l'avvio della sessione ma prima del controllo dei permessi.
 */
if ($_SERVER['PHP_SELF'] === $src_subdir . '/cliente/carrello.php') {
    if (isset($_POST['azione']) && $_POST['azione'] === 'aggiungi') {
        $_SESSION['agg_carr_id_prod'] = $_POST['id_prodotto'];
        $_SESSION['agg_carr_qta'] = $_POST['quantita'];
    }
}

$loggato = isset($_SESSION['id_utente']) && !is_nan($_SESSION['id_utente']);

$e_cliente = isset($_SESSION['tipo_utente']) && $_SESSION['tipo_utente'] === 'cli
ente';
$e_gestore = isset($_SESSION['tipo_utente']) && $_SESSION['tipo_utente'] === 'ges
tore';
$e_admin =    isset($_SESSION['tipo_utente']) && $_SESSION['tipo_utente'] === 'adm
in';
$e_visitatore = !$e_cliente && !$e_gestore && !$e_admin;

if ($e_visitatore && !$perm_visitatore) {
    redirect(401, $src_subdir . '/utente/login.php', true);
}

if (
    ($e_cliente && !$perm_cliente) ||
    ($e_gestore && !$perm_gestore) ||
    ($e_admin && !$perm_admin)
) {
    redirect(403, $src_subdir . '/accesso_negato.php', false);
}
?>
```

```
<?php
require_once($src_root . '/lib/constants.php');
require_once($src_root . '/lib/xml.php');

$doc_utenti = load_xml('utenti');

function login_utente($email, $password) {
    global $doc_utenti;

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@email='$email']")
;
    if ($result->length !== 1) {
        return "Errore email";
    }

    $utente = $result[0];

    $password2 = $utente->getElementsByTagName('password')[0]->textContent;
    if ($password2 !== md5($password)) {
        return "Errore password";
    }

    $tipo = $utente->getAttribute('tipo');

    if ($tipo === 'cliente') {
        $attivo = $utente->getElementsByTagName('attivo')[0]->textContent;
        if ($attivo !== 'true') {
            return "Errore abilitazione";
        }
    }

    $_SESSION['id_utente'] = $utente->getAttribute('id');
    $_SESSION['tipo_utente'] = $tipo;

    return "Ok";
}

function registra_utente($nome, $cognome, $email, $password, $telefono, $indirizz
o, $codice_fiscale) {
    global $doc_utenti;

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@email='$email']")
;
    if ($result->length !== 0) {
        return false;
    }

    $root = $doc_utenti->documentElement;
    $utenti = $root->childNodes;

    $nuovo_utente = $doc_utenti->createElement('utente');

    $id_utente = get_next_id($utenti);
    $nuovo_utente->setAttribute('id', $id_utente);
    $nuovo_utente->setAttribute('tipo', 'cliente');
    $nuovo_utente->setAttribute('email', $email);

    $el_attivo = $doc_utenti->createElement('attivo', 'true');
    $nuovo_utente->appendChild($el_attivo);

    $el_nome = $doc_utenti->createElement('nome', $nome);
    $nuovo_utente->appendChild($el_nome);

    $el_cognome = $doc_utenti->createElement('cognome', $cognome);
```

```
$nuovo_utente->appendChild($el_cognome);

$el_telefono = $doc_utenti->createElement('telefono', $telefono);
$nuovo_utente->appendChild($el_telefono);

$el_indirizzo = $doc_utenti->createElement('indirizzo', $indirizzo);
$nuovo_utente->appendChild($el_indirizzo);

$el_codice_fiscale = $doc_utenti->createElement('codiceFiscale', $codice_fiscal
e);
$nuovo_utente->appendChild($el_codice_fiscale);

$el_credito = $doc_utenti->createElement('credito', '0.00');
$nuovo_utente->appendChild($el_credito);

$el_reputazione = $doc_utenti->createElement('reputazione', '30');
$nuovo_utente->appendChild($el_reputazione);

$el_password = $doc_utenti->createElement('password', md5($password));
$nuovo_utente->appendChild($el_password);

$el_data_reg = $doc_utenti->createElement('dataRegistrazione', date('Y-m-d'));
$nuovo_utente->appendChild($el_data_reg);

$el_carrello = $doc_utenti->createElement('carrello');
$nuovo_utente->appendChild($el_carrello);

$root->appendChild($nuovo_utente);

save_xml($doc_utenti, 'utenti');

// BUG: se non ricarico il documento i prossimi xpath() continuano ad usare il
documento vecchio
$doc_utenti = load_xml('utenti');

return true;
}

function modifica_cliente($id_utente, $attivo, $nome, $cognome, $email, $password
, $telefono, $indirizzo, $codice_fiscale) {
    global $doc_utenti;

    $successo = true;

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id=$id_utente]");
    $utente = $result[0];

    $utente->getElementsByTagName('attivo')[0]->textContent = $attivo ? 'true' : 'f
alse';
    $utente->getElementsByTagName('nome')[0]->textContent = $nome;
    $utente->getElementsByTagName('cognome')[0]->textContent = $cognome;

    if (preg_match(REGEX_EMAIL, $email)) {
        $utente->setAttribute('email', $email);
        $successo = false;
    }

    if ($password !== '') {
        $utente->getElementsByTagName('password')[0]->textContent = md5($password);
        $successo = false;
    }

    if (preg_match(REGEX_TELEFONO, $telefono)) {
        $utente->getElementsByTagName('telefono')[0]->textContent = $telefono;
```

```
$successo = false;
}

if (preg_match(REGEX_INDIRIZZO, $indirizzo)) {
    $utente->getElementsByTagName('indirizzo')[0]->textContent = $indirizzo;
    $successo = false;
}

if (preg_match(REGEX_CF, $codice_fiscale)) {
    $utente->getElementsByTagName('codiceFiscale')[0]->textContent = $codice_fisc
ale;
    $successo = false;
}

save_xml($doc_utenti, 'utenti');

return $successo;
}

function modifica_ut_spec($id_utente, $nome, $cognome, $email, $password) {
    global $doc_utenti;

    $successo = true;

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id=$id_utente]");
    $utente = $result[0];

    $utente->getElementsByTagName('nome')[0]->textContent = $nome;
    $utente->getElementsByTagName('cognome')[0]->textContent = $cognome;

    if (preg_match(REGEX_EMAIL, $email)) {
        $utente->setAttribute('email', $email);
        $successo = false;
    }

    if ($password !== '') {
        $utente->getElementsByTagName('password')[0]->textContent = md5($password);
        $successo = false;
    }

    save_xml($doc_utenti, 'utenti');

    return $successo;
}

function ottieni_info_utente($id_utente) {
    global $doc_utenti;

    $result = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id='$id_utente']"
);
    $utente = $result[0];

    $tipo = $utente->getAttribute('tipo');

    $nome = $utente->getElementsByTagName('nome')[0]->textContent;
    $cognome = $utente->getElementsByTagName('cognome')[0]->textContent;
    $email = $utente->getAttribute('email');

    if ($tipo === 'gestore' || $tipo === 'admin') {
        return [
            'nome' => $nome,
            'cognome' => $cognome,
            'email' => $email
        ];
    }
}
```



```
}

$telefono = $utente->getElementsByTagName('telefono')[0]->textContent;
$indirizzo = $utente->getElementsByTagName('indirizzo')[0]->textContent;
$codice_fiscale = $utente->getElementsByTagName('codiceFiscale')[0]->textContent;
t;
$credito = $utente->getElementsByTagName('credito')[0]->textContent;
$reputazione = $utente->getElementsByTagName('reputazione')[0]->textContent;
$data_reg = $utente->getElementsByTagName('dataRegistrazione')[0]->textContent;

return [
    'nome' => $nome,
    'cognome' => $cognome,
    'email' => $email,
    'telefono' => $telefono,
    'indirizzo' => $indirizzo,
    'codice_fiscale' => $codice_fiscale,
    'credito' => $credito,
    'reputazione' => $reputazione,
    'data_reg' => $data_reg
];
}

function modifica_credito($diff) {
    global $doc_utenti;

    $id_utente = $_SESSION['id_utente'];

    $credito = xpath($doc_utenti, 'utenti', "/ns:utenti/ns:utente[@id='$id_utente']
/ns:credito")[0];
    $credito->textContent += $diff;

    save_xml($doc_utenti, 'utenti');

    return true;
}

function trova_gestore($id) {
    global $doc_utenti;

    $result = xpath($doc_utenti, 'utenti', '/ns:utenti/ns:utente[@id=' . $id . ' and
@tipo="gestore"]');

    return $result->length > 0;
}
?>
```

```
<?php
function redirect($code, $page, $use_dest) {
    if ($use_dest) {
        $cur_page = $_SERVER['REQUEST_URI'];
        $suffix = '?redirect=' . $cur_page;
    } else {
        $suffix = '';
    }

    http_response_code($code);
    header('Location: ' . $page . $suffix);
    exit();
}

function make_subdir($level) {
    $path = $_SERVER['PHP_SELF'];

    for ($i = 0; $i < ($level + 1); $i++) {
        $pos = strrpos($path, '/');
        $path = substr($path, 0, $pos);
    }

    return $path;
}
?>
```

```
<?php
function load_xml($table) {
    global $src_root;

    $xmlFile = $src_root . '/data/' . $table . '.xml';

    $xmlData = file($xmlFile);
    if (!$xmlData) {
        return null;
    }

    $xmlString = '';
    foreach($xmlData as $line) {
        $xmlString .= trim($line);
    }
    $doc = new DOMDocument();
    $doc->loadXML($xmlString);

    return $doc;
}

function save_xml($doc, $table) {
    global $src_root;

    $xmlFile = $src_root . '/data/' . $table . '.xml';

    $doc->formatOutput = true;
    $doc->preserveWhiteSpace = false;

    $doc->save($xmlFile);
}

function domlist_to_array($domlist) {
    $arr = [];
    for ($i = 0; $i < $domlist->length; $i++) {
        $arr[$i] = $domlist->item($i);
    }
    return $arr;
}

function sort_by_element_dec($elements, $key, $desc) {
    usort($elements, function($aElement, $bElement) use ($key, $desc) {
        $aValue = $aElement->getElementsByTagName($key)[0]->textContent;
        $bValue = $bElement->getElementsByTagName($key)[0]->textContent;

        $aValue = floatval($aValue);
        $bValue = floatval($bValue);

        if ($desc) {
            return $bValue <=> $aValue;
        } else {
            return $aValue <=> $bValue;
        }
    });

    return $elements;
}

function sort_by_element_txt($elements, $key, $desc) {
    usort($elements, function($aElement, $bElement) use ($key, $desc) {
        $aValue = $aElement->getElementsByTagName($key)[0]->textContent;
        $bValue = $bElement->getElementsByTagName($key)[0]->textContent;

        if ($desc) {
```

```
        return strnatcmp($bValue, $aValue);
    } else {
        return strnatcmp($aValue, $bValue);
    }
});

return $elements;
}

function get_next_id($elements) {
    $next_id = 0;
    foreach ($elements as $element) {
        $cur_id = $element->getAttribute('id');
        if ($next_id < $cur_id) {
            $next_id = $cur_id;
        }
    }
    $next_id++;

    return $next_id;
}

function xpath($doc, $table, $query) {
    $xp = new DOMXPath($doc);
    $xp->registerNamespace('ns', 'http://www.lweb.uni/tesina-rcstore/' . $table . '
/');
    $result = $xp->evaluate($query);
    return $result;
}
?>
```