

# IWT Lab Project Report

## Project Credits:

- Dhriti Sojitra (23BCP341)
- Nevil Shah (23BCP345)

## Project Submitted to faculty in charge (G10):

- Dr. Punit Gupta

## GitHub Link:

[https://github.com/NevilShah26/currency\\_tax\\_estimator.git](https://github.com/NevilShah26/currency_tax_estimator.git)

## Project Aim:

Estimating Duty Charges for Import trade

## Tech Stack:

- Frontend: HTML, CSS, JS
- Backend: PHP, FPDF library
- Server: Apache & MySQL via XAMPP
- IDE: VS Code

## Project Structure:

```
∨ fpdf186
  > doc
  > font
  > makefont
  > tutorial
  ⌂ changelog.htm
  ⌂ FAQ.htm
  # fpdf.css
  🐘 fpdf.php
  ≡ install.txt
  🐘 license.txt
  🐘 api.php
  🐘 generate_pdf.php
  ⌂ index.html
  🐘 receipt.php
  # styles.css
```

## Source Code:

### 1. api.php

```
<?php

header('Content-Type: application/json; charset=utf-8');

if (isset($_SERVER['HTTP_ORIGIN']) && ($_SERVER['HTTP_ORIGIN'] !== '')) {
    header('Access-Control-Allow-Origin: ' . $_SERVER['HTTP_ORIGIN']);
    header('Access-Control-Allow-Headers: Content-Type');
}

$raw = file_get_contents('php://input');
if (!$raw) { http_response_code(400); echo json_encode(['error'=>'No input received']); exit; }

$input = json_decode($raw, true);
if (!$input) { http_response_code(400); echo json_encode(['error'=>'Invalid JSON']); exit; }

$amount = floatval($input['amount'] ?? 0);
$from = strtoupper(trim($input['from'] ?? 'USD'));
$to = strtoupper(trim($input['to'] ?? 'INR'));
$category = trim($input['category'] ?? 'electronics');

if ($amount <= 0) {
    http_response_code(400);
    echo json_encode(['error'=>'Amount must be > 0']);
    exit;
}

try {

    $rate = null;
    $url = "https://open.er-api.com/v6/latest/{$from}";
    $resp = @file_get_contents($url);

    if ($resp !== false) {
        $data = json_decode($resp, true);
    }
}
```

```
        if      (isset($data['rates'][$to]))    $rate    =
floatval($data['rates'][$to]);
    }

    if  (!$rate) {
        $apiKey = "b3a716b33fa7ca32c5ebe9210deae46d";
$url2      =
"https://api.currencyfreaks.com/latest?apikey={$apiKey}&symbols={$to}&base={$from}";
        $resp2 = @file_get_contents($url2);
        if  ($resp2 !== false) {
            $data2 = json_decode($resp2, true);
                if      (isset($data2['rates'][$to]))    $rate    =
floatval($data2['rates'][$to]);
            }
        }
    }

    if  (!$rate || $rate <= 0) {
        throw new Exception('Rate not present in API response');
    }

$converted_price = $amount * $rate;

$subType = 'any';

if  ($category === "electronics") {
    $subType = $input['electronicsType'] ?? 'standard';
} elseif ($category === "books") {
    $subType = $input['bookType'] ?? 'printed';
} elseif ($category === "beauty_products") {
    $subType = $input['beautyType'] ?? 'cosmetic';
}

if  ($category === "clothing") {
    $priceCondition = ($converted_price <= 1000) ? "<=1000" : ">1000";
} else {
    $priceCondition = "any";
}

$pdo = new PDO(
    'mysql:host=localhost;dbname=mysql;charset=utf8mb4',
    'root',
```

```

    ''',
    [ PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
);

$stmt = $pdo->prepare("
    SELECT * FROM tax_rules
    WHERE category = ? AND sub_type = ? AND price_condition = ?
    LIMIT 1
");
$stmt->execute([$category, $subType, $priceCondition]);
$rule = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$rule) {
    $stmt = $pdo->prepare("SELECT * FROM tax_rules WHERE category =
? LIMIT 1");
    $stmt->execute([$category]);
    $rule = $stmt->fetch(PDO::FETCH_ASSOC);
}

if (!$rule) throw new Exception("No tax rule found in DB");

$gst = $rule['gst_rate'];
$bcd = $rule['bcd_rate'];
$sws = $rule['sws_rate'];
$cess = $rule['cess_rate'];

$bcd_amt = $converted_price * ($bcd / 100);
$sws_amt = $bcd_amt * ($sws / 100);
$gst_amt = ($converted_price + $bcd_amt + $sws_amt) * ($gst / 100);
$cess_amt = ($cess > 0) ? ($converted_price + $bcd_amt) * ($cess / 100) :
0;

$total = $converted_price + $bcd_amt + $sws_amt + $gst_amt + $cess_amt;

$fmt = fn($v) => number_format(round($v, 2), 2);

echo json_encode([
    'rate' => $rate,
    'converted_price' => $converted_price,
    'converted_price_formatted' => "₹". $fmt($converted_price),

    'customs_rate' => $bcd,
    'customs' => $bcd_amt,
]);

```

```

'customs_formatted' => "₹".$fmt($bcd_amt) ,

'sws_rate' => $sws,
'sws' => $sws_amt,
'sws_formatted' => "₹".$fmt($sws_amt) ,

'gst_rate' => $gst,
'gst' => $gst_amt,
'gst_formatted' => "₹".$fmt($gst_amt) ,

'cess_rate' => $cess,
'cess' => $cess_amt,
'cess_formatted' => $cess_amt ? "₹".$fmt($cess_amt) : null ,

'total' => $total,
'total_formatted' => "₹".$fmt($total)
]) ;

} catch (Exception $e) {
    http_response_code(500);
    echo json_encode(['error'=>$e->getMessage()]);
}
?>

```

## 2. generate\_pdf.php

```

<?php
ob_clean();
error_reporting(0);
ini_set('display_errors', 0);

require('fpdf186/fpdf.php');

$data = json_decode(file_get_contents("php://input"), true);
if(!$data) {
    http_response_code(400);
    echo "Invalid data";
    exit;
}

header('Content-Type: application/pdf');
header('Content-Disposition: attachment; filename="receipt.pdf"');

function cleanSymbol($text) {

```

```

    return str_replace('₹', 'Rs. ', $text);
}

$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(0,10,'Import Duty & Tax Calculation Receipt',0,1,'C');
$pdf->Ln(10);

$pdf->SetFont('Arial','',12);
$pdf->Cell(0,8,'Date: '.date('d M Y, H:i:s'),0,1,'R');
$pdf->Cell(0,8,'Category: '.ucfirst($data['category']),0,1,'R');
$pdf->Ln(5);

$pdf->SetFont('Arial','B',12);
$pdf->SetFillColor(240, 240, 240);
$pdf->Cell(90,10,'Detail',1,0,'C',true);
$pdf->Cell(90,10,'Amount / Rate',1,1,'C',true);

$pdf->SetFont('Arial','',12);
function TableRow($label, $value) {
    global $pdf;
    $pdf->Cell(90,10,$label,1,0,'L');
    $pdf->Cell(90,10,$value,1,1,'R');
}

TableRow('Product Value', $data['amount'].' '.$data['from']);
TableRow('Exchange Rate', '1 '.$data['from'].' = '.$data['rate'].' '.$data['to']);
TableRow('Converted', 'Value', cleanSymbol($data['converted_price_formatted']));
TableRow('BCD', ('.'.$data['customs_rate'].'%'),
cleanSymbol($data['customs_formatted']));
TableRow('SWS', ('.'.$data['sws_rate'].'%'),
cleanSymbol($data['sws_formatted']));
TableRow('GST', ('.'.$data['gst_rate'].'%'),
cleanSymbol($data['gst_formatted']));

if(isset($data['cess_formatted']) && $data['cess_formatted'])
    TableRow('Cess', ('.'.$data['cess_rate'].'%'),
cleanSymbol($data['cess_formatted']));
else
    TableRow('Cess', 'N/A');

```

```

$pdf->SetFont('Arial','B',13);
$pdf->SetFillColor(255, 234, 200);
$pdf->Cell(90,10,'Total Landed Cost',1,0,'L',true);
$pdf->Cell(90,10,cleanSymbol($data['total_formatted']),1,1,'R',true);

$pdf->Ln(10);
$pdf->SetFont('Arial','I',10);
$pdf->Cell(0,10,'This is a system-generated receipt based on import duty
rules.',0,1,'C');

$pdf->Output('I', 'receipt.pdf');
exit;
?>

```

### 3. index.html

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Import Cost & Duty Calculator</title>
<link rel="stylesheet" href="style.css">
</head>
<body>

<div class="card">
<h2>Import Cost & Duty Calculator</h2>

<form id="estimator">

<label>Currency amount
<input type="number" step="0.01" id="amount" name="amount"
value="100" required />
</label>

<label>Currency (from)
<select id="currency" name="currency">
<option value="USD">USD</option>
<option value="EUR">EUR</option>
<option value="GBP">GBP</option>
<option value="AED">AED</option>

```

```

        </select>
    </label>

    <label>Category
        <select id="category" name="category">
            <option value="electronics">Electronics</option>
            <option value="clothing">Clothing</option>
            <option value="books">Books</option>
            <option value="beauty_products">Beauty Products</option>
        </select>
    </label>

    <div id="dynamic-fields"></div>

    <label>Ship to (target currency)
        <select id="target" name="target">
            <option value="INR">INR</option>
        </select>
    </label>

    <button type="submit">Calculate</button>
</form>

<div id="output" class="result" hidden></div>
<button id="downloadPDF" hidden>Download PDF Receipt</button>

<h4>Notes</h4>
<ul>
    <li>Rates & duties vary by product type. Please select correct category details.</li>
        <li>Exchange rate is fetched from API & may differ from live forex rate.</li>
        <li>
            Taxes and duty rates applied are based on the latest GST Council and Customs revisions
            effective from <strong>22 September 2025</strong>, including updated slabs for electronics,
            clothing, beauty products, and books.
        </li>
</ul>
</div>

<script>
const form = document.getElementById('estimator');

```

```
const output = document.getElementById('output');
const downloadBtn = document.getElementById('downloadPDF');
const categorySelect = document.getElementById('category');
const dynamicFields = document.getElementById('dynamic-fields');
let lastResult = null;

function updateDynamicFields() {
    const cat = categorySelect.value;
    dynamicFields.innerHTML = "";

    if (cat === "electronics") {
        dynamicFields.innerHTML = `
            <label>Electronics Type
            <select id="electronicsType">
                <option value="standard">Standard Electronics (18% GST)</option>
                <option value="luxury">Premium / Luxury Electronics (28% GST)</option>
            </select>
        </label>
    `;
    }

    if (cat === "books") {
        dynamicFields.innerHTML = `
            <label>Book Type
            <select id="bookType">
                <option value="printed">Printed Book (0% GST)</option>
                <option value="ebook">E-Book (18% GST)</option>
            </select>
        </label>
    `;
    }

    if (cat === "beauty_products") {
        dynamicFields.innerHTML = `
            <label>Beauty Product Type
            <select id="beautyType">
                <option value="essential">Essential Toiletry (5% GST)</option>
                <option value="cosmetic">Standard Cosmetic (18% GST)</option>
                <option value="luxury">Luxury Perfume / Cosmetic (28% GST)</option>
            </select>
        </label>
    `;
    }
}
```

```
}

categorySelect.addEventListener("change", updateDynamicFields);
updateDynamicFields();

form.addEventListener('submit', async (e) => {
  e.preventDefault();
  output.hidden = true;
  downloadBtn.hidden = true;

  const data = {
    amount: parseFloat(document.getElementById('amount').value),
    from: document.getElementById('currency').value,
    to: document.getElementById('target').value,
    category: document.getElementById('category').value
  };

  if (data.category === "electronics")
    data.electronicsType = document.getElementById("electronicsType").value;

  if (data.category === "books")
    data.bookType = document.getElementById("bookType").value;

  if (data.category === "beauty_products")
    data.beautyType = document.getElementById("beautyType").value;

  output.textContent = 'Calculating...';
  output.hidden = false;

  try {
    const res = await fetch('api.php', {
      method: 'POST',
      headers: {'Content-Type': 'application/json'},
      body: JSON.stringify(data)
    });

    const json = await res.json();
    if (!res.ok) throw new Error(json.error || 'Server error');

    lastResult = { ...json, ...data };
    output.innerHTML =
      <strong>Exchange Rate:</strong> 1 ${data.from} = ${json.rate}
${data.to}<br>
```

```

        <strong>Converted Value:</strong> ${json.converted_price_formatted}<br>
            <strong>Customs Duty (${json.customs_rate}%)</strong>
${json.customs_formatted}<br>
        <strong>GST (${json.gst_rate}%)</strong> ${json.gst_formatted}<br>
        ${json.cess ? `<strong>Cess ($${json.cess_rate}%)</strong>
${json.cess_formatted}<br>` : ''}
<hr><strong>Total Landed Cost:</strong> ${json.total_formatted}
`;

downloadBtn.hidden = false;

} catch (err) {
    output.textContent = 'Error: ' + err.message;
}
}) ;

downloadBtn.addEventListener('click', async () => {
if (!lastResult) return alert("Please calculate first!");
const res = await fetch('generate_pdf.php', {
    method: 'POST',
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify(lastResult)
});
const blob = await res.blob();
const url = URL.createObjectURL(blob);
const a = document.createElement('a');
a.href = url;
a.download = 'receipt.pdf';
a.click();
URL.revokeObjectURL(url);
});
</script>
</body>
</html>

```

#### 4. styles.css

```

@import
url('https://fonts.googleapis.com/css2?family=Roboto+Mono:wght@400;600;700&display=swap');

:root {
--color-primary-dark: #000000;
--color-background: #F8F4E3;
--color-card-background: #ffffff;
--color-accent: #FF6B00;

```

```
--color-text-dark: #000000;
--color-secondary-accent: #60B7FF;
--border-weight: 4px;
--shadow-offset: 8px;
}

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {

  font-family: 'Roboto Mono', monospace;
  background-color: var(--color-background);
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  padding: 20px;
  color: var(--color-text-dark);
}

h2 {
  font-weight: 700;
  color: var(--color-text-dark);
  margin-bottom: 1.5rem;
  font-size: 2rem;
  text-transform: uppercase;
  letter-spacing: -1px;
}

h4 {
  font-weight: 700;
  color: var(--color-text-dark);
  margin-top: 2rem;
  margin-bottom: 0.5rem;
  font-size: 1.1rem;
}

ul {
  list-style: square inside;
  padding-left: 1.2rem;
  margin: 0;
```

```
font-size: 0.9rem;
color: var(--color-text-dark);
}

li {
  line-height: 1.6;
}

.card {
  background: var(--color-card-background);
  border: var(--border-weight) solid var(--color-primary-dark);
  border-radius: 0;
  padding: 2.5rem;
  width: 100%;
  max-width: 500px;
  box-shadow: var(--shadow-offset) var(--color-primary-dark) 0 0 0 0;
  transition: transform 0.3s ease-in-out;
  animation: slideIn 0.5s ease-out;
}

@keyframes slideIn {
  from {
    opacity: 0;
    transform: translateX(-50px);
  }
  to {
    opacity: 1;
    transform: translateX(0);
  }
}

#estimator {
  display: grid;
  gap: 1.5rem;
}

label {
  display: flex;
  flex-direction: column;
  font-weight: 600;
  color: var(--color-text-dark);
  font-size: 0.95rem;
  text-transform: uppercase;
}
```

```
}

input[type="number"],
select {
  width: 100%;
  padding: 0.75rem 1rem;
  margin-top: 0.5rem;
  border: 2px solid var(--color-primary-dark);
  border-radius: 0;
  background-color: var(--color-background);
  font-family: inherit;
  font-size: 1rem;
  color: var(--color-text-dark);
  appearance: none;
  -webkit-appearance: none;
  -moz-appearance: none;
  transition: background-color 0.2s;
}

label:has(select) {
  position: relative;
}

label:has(select)::after {
  content: '▼';
  position: absolute;
  top: 60%;
  right: 15px;
  pointer-events: none;
  color: var(--color-primary-dark);
  font-size: 0.75rem;
  font-weight: 700;
}

input[type="number"]:focus,
select:focus {
  outline: none;
  border-color: var(--color-accent);
  background-color: #FFFCEF;
}

button {
  padding: 0.85rem 1.5rem;
  border: var(--border-weight) solid var(--color-primary-dark);
  border-radius: 0;
```

```
font-size: 1.1rem;
font-weight: 700;
color: var(--color-primary-dark);
cursor: pointer;
text-transform: uppercase;
box-shadow: 4px 4px 0 var(--color-primary-dark);
transition: all 0.2s ease-out;
margin-top: 1rem;
}

button[type="submit"] {
background-color: var(--color-accent);
}

#downloadPDF {
background-color: var(--color-secondary-accent);
margin-top: 1rem;
width: 100%;
}

button:hover {
transform: translate(4px, 4px);
box-shadow: 0 0 0 var(--color-primary-dark);
}

button[type="submit"]:hover {
background-color: #FF8A3D;
}

#downloadPDF:hover {
background-color: #79C2FF; }

button:active {
transform: translate(6px, 6px);
}

.result {
margin-top: 1.5rem;
padding: 1rem 1.5rem;
border: var(--border-weight) dashed var(--color-accent);
background-color: var(--color-card-background);
border-radius: 0;
font-size: 1.1rem;
```

```
font-weight: 700;
color: var(--color-text-dark);
line-height: 1.5;
}

.result hr {
border: none;
border-top: 2px solid var(--color-primary-dark);
margin: 0.75rem 0;
}

@media (max-width: 600px) {
.card {
padding: 1.5rem;
}
h2 {
font-size: 1.5rem;
margin-bottom: 1rem;
}
button {
font-size: 1rem;
}
:root {
--border-weight: 3px;
--shadow-offset: 6px;
}
}
```

## **Output:**

**Enter currency amount and select product category:**

### IMPORT COST & DUTY CALCULATOR

CURRENCY AMOUNT

CURRENCY (FROM)

CATEGORY

ELECTRONICS TYPE

## **Select Sub-Category:**

ELECTRONICS TYPE

CALCULATE

**Notes**

- Rates & duties vary by product type. Please select correct category details.
- Exchange rate is fetched from API & may differ from live forex rate.
- Taxes and duty rates applied are based on the latest GST Council and Customs revisions effective from 22 September 2025, including updated slabs for electronics, clothing, beauty products, and books.

## Calculated import cost with custom duty and other taxes:

INR

CALCULATE

Exchange Rate: 1 USD = 88.721738 INR  
Converted Value: ₹8,872.17  
Customs Duty (20.00%): ₹1,774.43  
GST (28.00%): ₹3,030.73  

---

Total Landed Cost: ₹13,854.79

DOWNLOAD PDF RECEIPT

**Notes**  
■ Rates & duties vary by product type. Please

## Downloaded Receipt:

### Import Duty & Tax Calculation Receipt

Date: 09 Nov 2025, 11:17:34

Category: Electronics

Detail	Amount / Rate
Product Value	100 USD
Exchange Rate	1 USD = 88.721738 INR
Converted Value	Rs. 8,872.17
BCD (20.00%)	Rs. 1,774.43
SWS (10.00%)	Rs. 177.44
GST (28.00%)	Rs. 3,030.73
Cess	N/A
<b>Total Landed Cost</b>	<b>Rs. 13,854.79</b>

## **Learnings:**

Through this project, we gained hands-on experience in both technical development and understanding real-world import taxation.

## **Technical Learnings:**

- Integrated live currency APIs and implemented fallback logic.
- Applied Indian import tax structure: **BCD, SWS, GST, and Cess**.
- Built dynamic backend tax logic with PHP and MySQL.
- Designed a flexible **tax\_rules** table for category-based and price-based tax rules.
- Developed a responsive UI and dynamic form handling using **HTML, CSS, and JavaScript**.
- Generated professional PDF receipts using FPDF with detailed tax breakup.
- Strengthened debugging, API handling, and server-side processing skills using **XAMPP**.

## **Domain Learnings:**

- Understood GST reforms effective from **22 September 2025**.
- Learned how classification (electronics, clothing, books, beauty products) affects tax rates.
- Implemented the logic for price-based GST applicability (e.g., clothing  $\leq$ ₹1000 vs  $>$ ₹1000).
- Understood real import cost components and calculation of **total landed cost**.

