MUNGAKHA A.C.K
PRIMARY & JUNIOR SCHOOL
P.O BOX 7-50205,
MUKHE VIA WEBUYE

Tonny Wanjala Mulati

ACADEMIC REPORTCARDS SOFTWARE NOTES

Prepared for schools

Automation Engineering

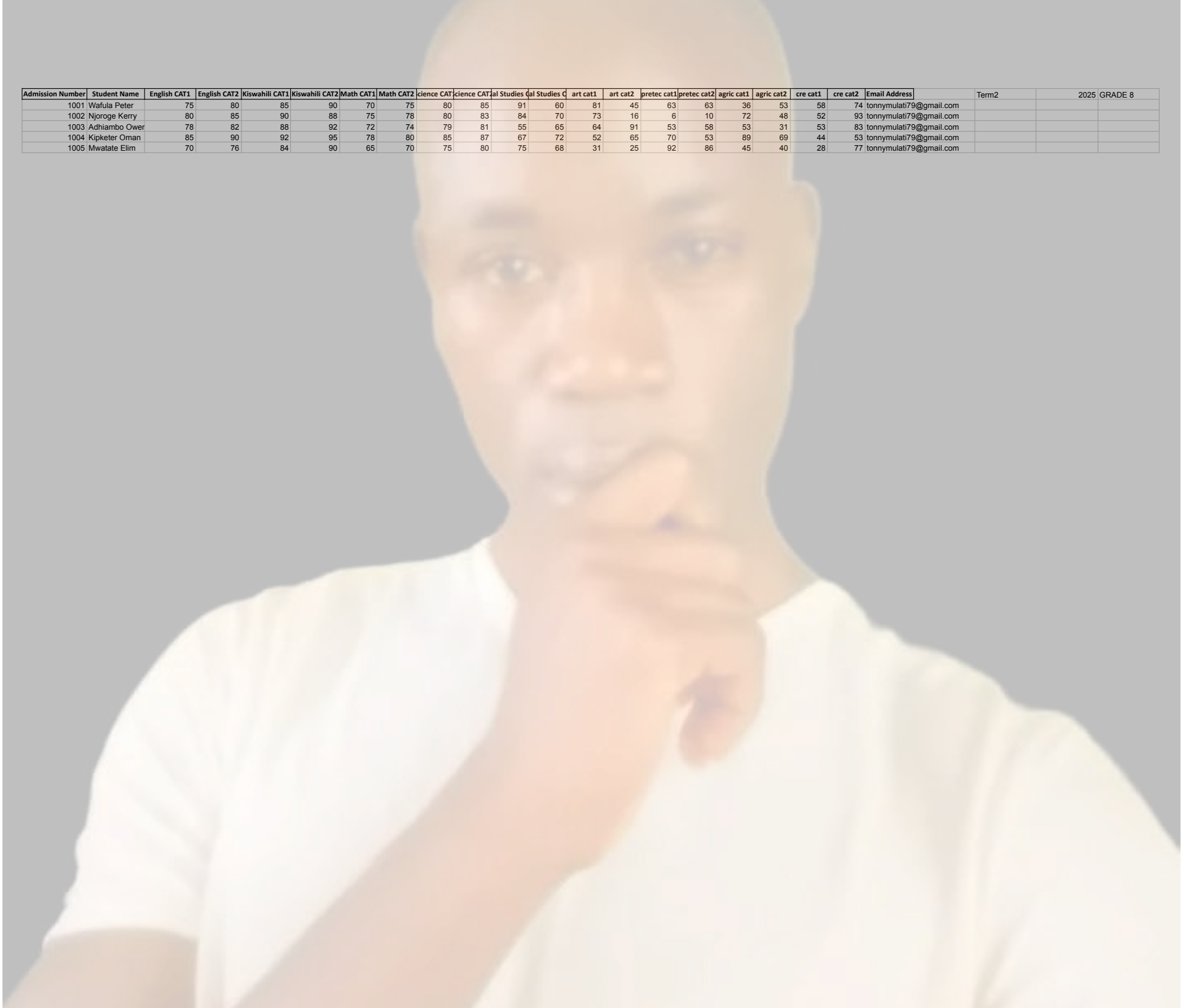
Created By

TONNY MULATI

+254703426340

tonnymulati79@gmail.com

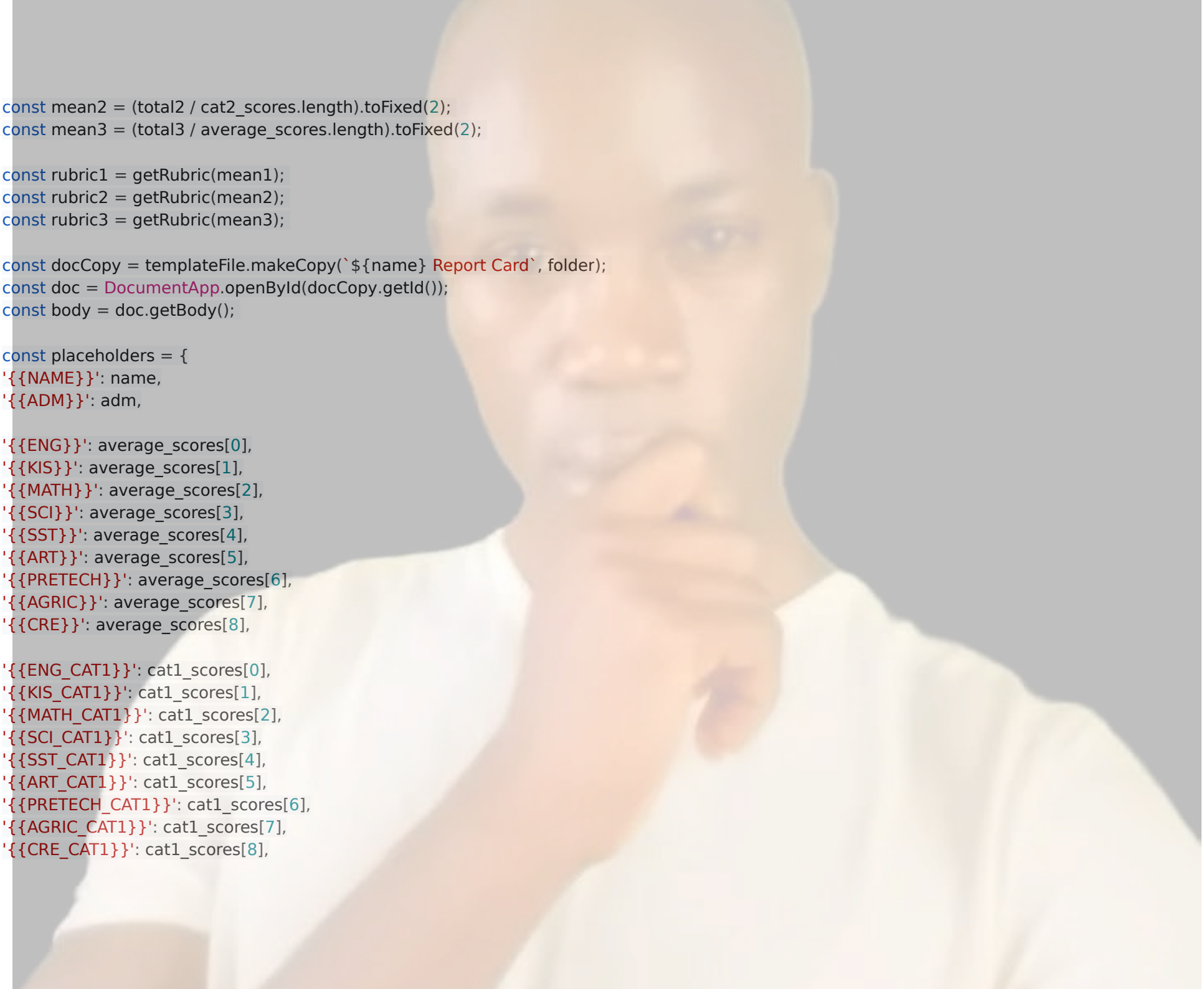
Admission Number	Student Name	English CAT1	English CAT2	Kiswahili CAT1	Kiswahili CAT2	Math CAT1	Math CAT2	cience CAT	cience CAT	al Studies C	al Studies C	art cat1	art cat2	pretec cat1	pretec cat2	agric cat1	agric cat2	cre cat1	cre cat2	Email Address	Term2	2025	GRADE 8
1001	Wafula Peter	75	80	85	90	70	75	80	85	91	60	81	45	63	63	36	53	58	74	tonnymulati79@gmail.com			
1002	Njoroge Kerry	80	85	90	88	75	78	80	83	84	70	73	16	6	10	72	48	52	93	tonnymulati79@gmail.com			
1003	Adhiambo Ower	78	82	88	92	72	74	79	81	55	65	64	91	53	58	53	31	53	83	tonnymulati79@gmail.com			
1004	Kipketer Oman	85	90	92	95	78	80	85	87	67	72	52	65	70	53	89	69	44	53	tonnymulati79@gmail.com			
1005	Mwatate Elim	70	76	84	90	65	70	75	80	75	68	31	25	92	86	45	40	28	77	tonnymulati79@gmail.com			



This project belongs to Tonny Wanjala Mulati. Email: tonnymulati79@gmail.com Call: +254703426340

Code.gs

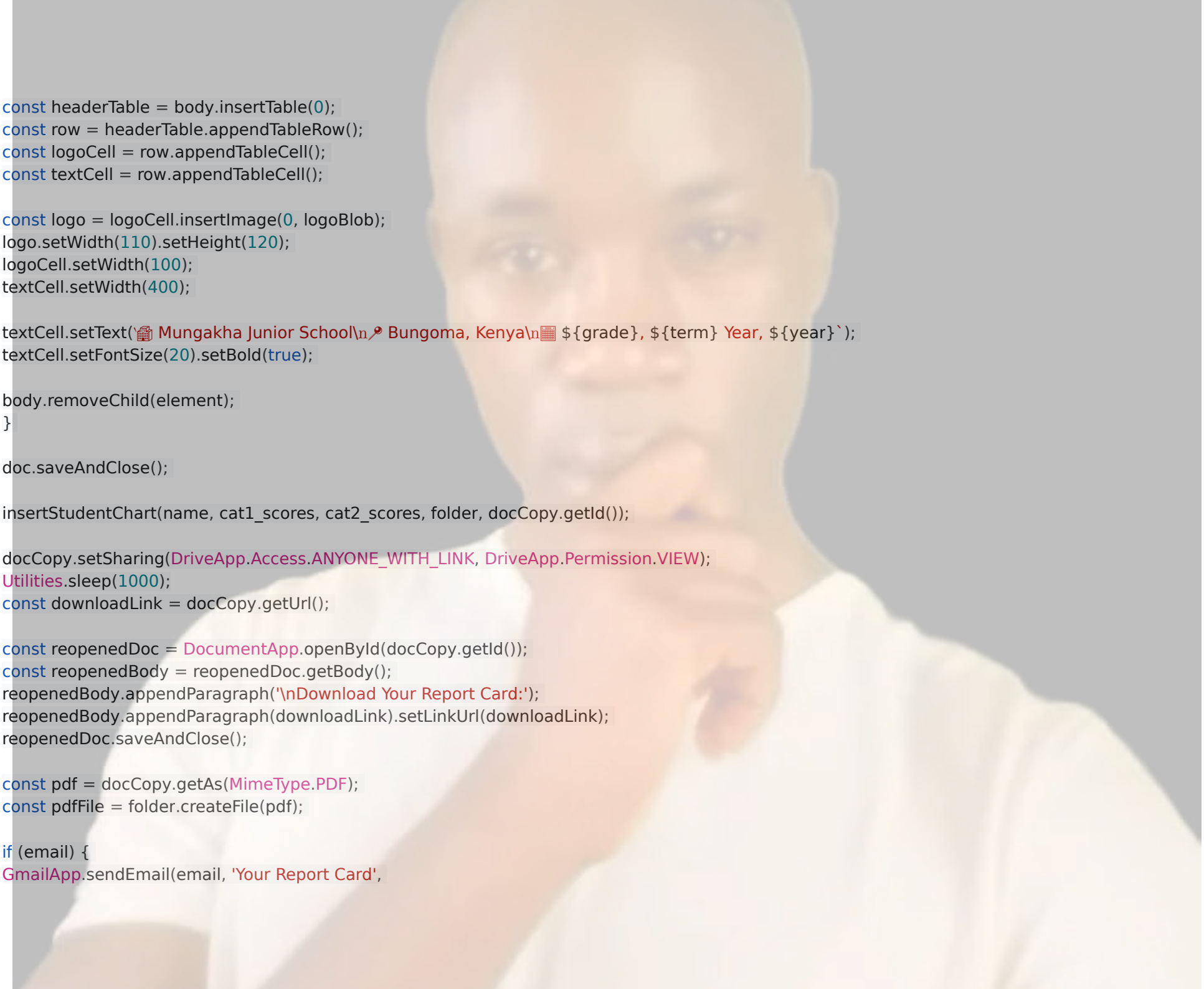
```
function generateReportCardsWithCharts() {
const ss = SpreadsheetApp.getActiveSpreadsheet();
const sheet = ss.getActiveSheet();
const data = sheet.getDataRange().getValues();
const templateFile = DriveApp.getFileById('1C2ZfZZCkP3Xlw8QP5afjNrH0eNgEp1CbnIeNyBwQsA4');
const folder = DriveApp.createFolder('Student Report Cards ' + new Date().toISOString());

const term = sheet.getRange("W1").getValue(); // Term
const year = sheet.getRange("X1").getValue(); // Year
const grade = sheet.getRange("Y1").getValue(); // Grade

const logoUrl = "https://drive.google.com/uc?export=view&id=1afqTLwfAhb3oGVC8J4v-tavj_J8sfdwE";
const logoBlob = UrlFetchApp.fetch(logoUrl).getBlob();

for (let i = 1; i < data.length; i++) {
try {
const [adm, name, eng_cat1, eng_cat2, kis_cat1, kis_cat2, math_cat1, math_cat2,
sci_cat1, sci_cat2, sst_cat1, sst_cat2,
art_cat1, art_cat2, pretech_cat1, pretech_cat2,
agric_cat1, agric_cat2, cre_cat1, cre_cat2,
email] = data[i];

const cat1_scores = [eng_cat1, kis_cat1, math_cat1, sci_cat1, sst_cat1, art_cat1, pretech_cat1, agric_cat1, cre_cat1].map(Number);
const cat2_scores = [eng_cat2, kis_cat2, math_cat2, sci_cat2, sst_cat2, art_cat2, pretech_cat2, agric_cat2, cre_cat2].map(Number);
const average_scores = cat1_scores.map((s1, idx) => ((s1 + cat2_scores[idx]) / 2).toFixed(2));

const total1 = cat1_scores.reduce((sum, val) => sum + val, 0);
const total2 = cat2_scores.reduce((sum, val) => sum + val, 0);
const total3 = average_scores.reduce((sum, val) => sum + Number(val), 0);

const mean1 = (total1 / cat1_scores.length).toFixed(2);
```



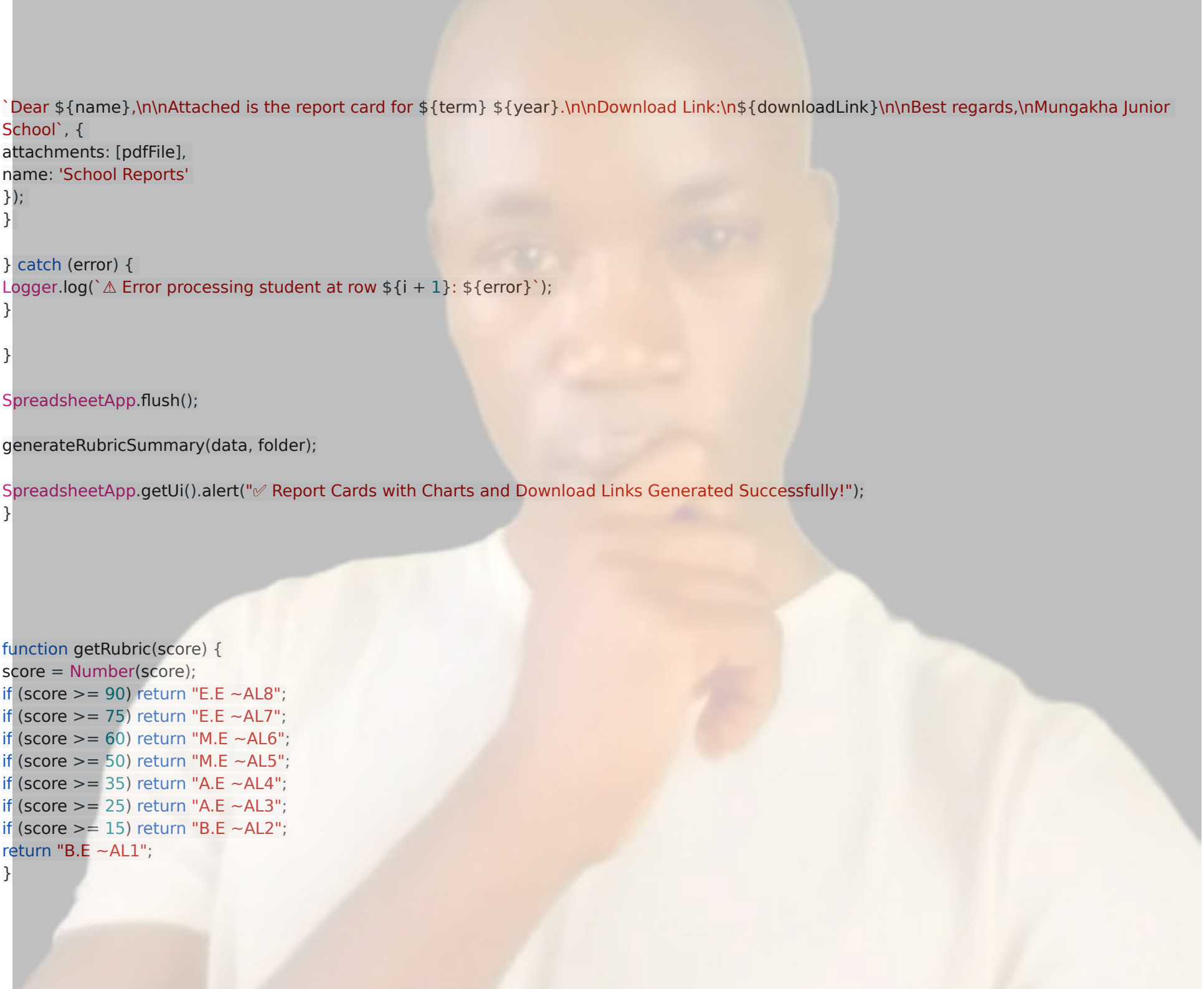
```
const mean2 = (total2 / cat2_scores.length).toFixed(2);
const mean3 = (total3 / average_scores.length).toFixed(2);

const rubric1 = getRubric(mean1);
const rubric2 = getRubric(mean2);
const rubric3 = getRubric(mean3);

const docCopy = templateFile.makeCopy(`${name} Report Card`, folder);
const doc = DocumentApp.openById(docCopy.getId());
const body = doc.getBody();

const placeholders = {
'{{NAME}}': name,
'{{ADM}}': adm,

'{{ENG}}': average_scores[0],
'{{KIS}}': average_scores[1],
'{{MATH}}': average_scores[2],
'{{SCI}}': average_scores[3],
'{{SST}}': average_scores[4],
'{{ART}}': average_scores[5],
'{{PRETECH}}': average_scores[6],
'{{AGRIC}}': average_scores[7],
'{{CRE}}': average_scores[8],

'{{ENG_CAT1}}': cat1_scores[0],
'{{KIS_CAT1}}': cat1_scores[1],
'{{MATH_CAT1}}': cat1_scores[2],
'{{SCI_CAT1}}': cat1_scores[3],
'{{SST_CAT1}}': cat1_scores[4],
'{{ART_CAT1}}': cat1_scores[5],
'{{PRETECH_CAT1}}': cat1_scores[6],
'{{AGRIC_CAT1}}': cat1_scores[7],
'{{CRE_CAT1}}': cat1_scores[8],
```

```
'{{ENG_CAT2}}': cat2_scores[0],
'{{KIS_CAT2}}': cat2_scores[1],
'{{MATH_CAT2}}': cat2_scores[2],
'{{SCI_CAT2}}': cat2_scores[3],
'{{SST_CAT2}}': cat2_scores[4],
'{{ART_CAT2}}': cat2_scores[5],
'{{PRETECH_CAT2}}': cat2_scores[6],
'{{AGRIC_CAT2}}': cat2_scores[7],
'{{CRE_CAT2}}': cat2_scores[8],

// CAT1 rubrics
'{{ENG_CAT1_RUBRIC}}': getRubric(cat1_scores[0]),
'{{KIS_CAT1_RUBRIC}}': getRubric(cat1_scores[1]),
'{{MATH_CAT1_RUBRIC}}': getRubric(cat1_scores[2]),
'{{SCI_CAT1_RUBRIC}}': getRubric(cat1_scores[3]),
'{{SST_CAT1_RUBRIC}}': getRubric(cat1_scores[4]),
'{{ART_CAT1_RUBRIC}}': getRubric(cat1_scores[5]),
'{{PRETECH_CAT1_RUBRIC}}': getRubric(cat1_scores[6]),
'{{AGRIC_CAT1_RUBRIC}}': getRubric(cat1_scores[7]),
'{{CRE_CAT1_RUBRIC}}': getRubric(cat1_scores[8]),

// CAT2 rubrics
'{{ENG_CAT2_RUBRIC}}': getRubric(cat2_scores[0]),
'{{KIS_CAT2_RUBRIC}}': getRubric(cat2_scores[1]),
'{{MATH_CAT2_RUBRIC}}': getRubric(cat2_scores[2]),
'{{SCI_CAT2_RUBRIC}}': getRubric(cat2_scores[3]),
'{{SST_CAT2_RUBRIC}}': getRubric(cat2_scores[4]),
'{{ART_CAT2_RUBRIC}}': getRubric(cat2_scores[5]),
'{{PRETECH_CAT2_RUBRIC}}': getRubric(cat2_scores[6]),
'{{AGRIC_CAT2_RUBRIC}}': getRubric(cat2_scores[7]),
'{{CRE_CAT2_RUBRIC}}': getRubric(cat2_scores[8]),

// Average rubrics
'{{ENG_RUBRIC}}': getRubric(average_scores[0]),
'{{KIS_RUBRIC}}': getRubric(average_scores[1]),
```

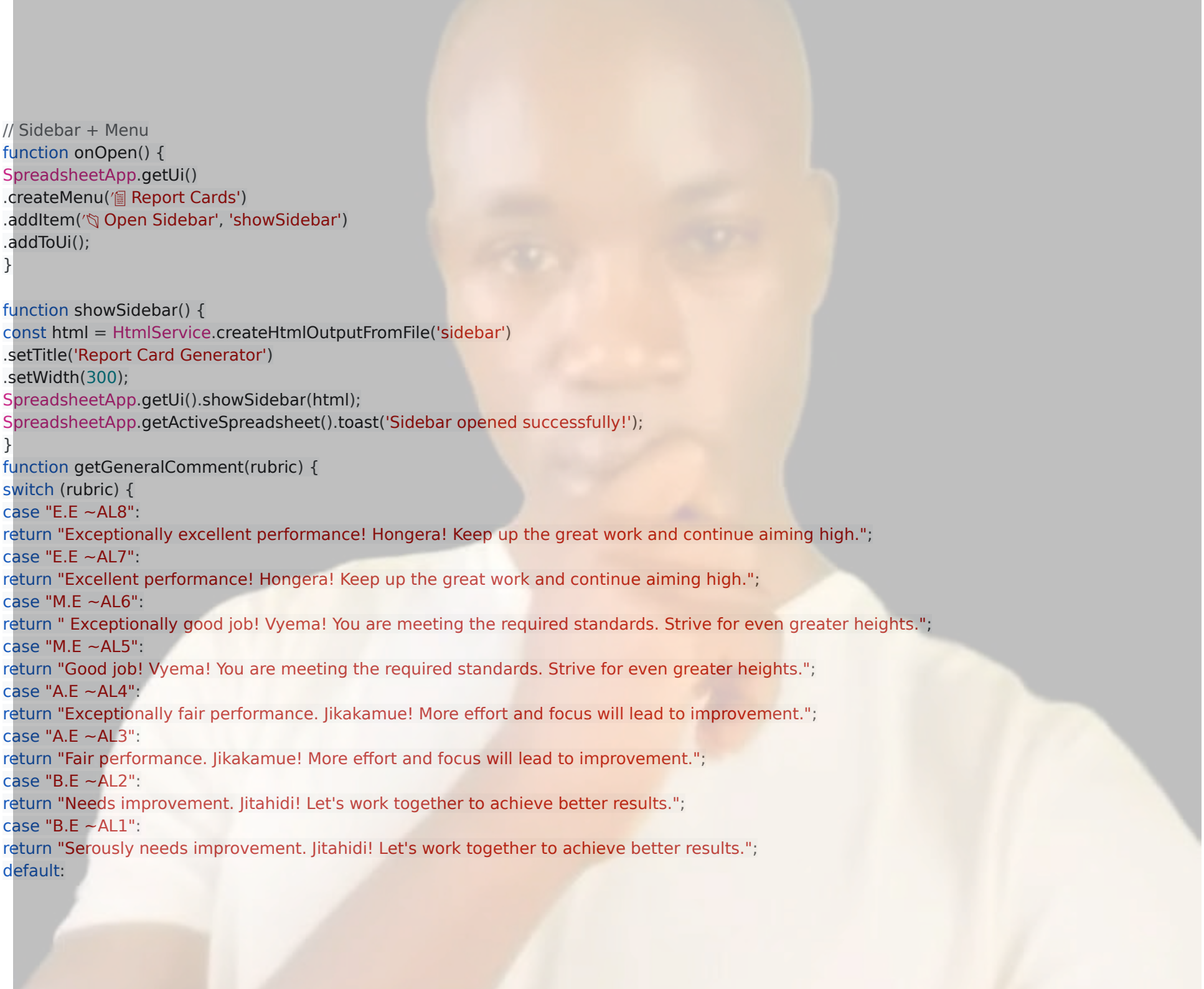
```
'{{MATH_RUBRIC}}': getRubric(average_scores[2]),
'{{SCI_RUBRIC}}': getRubric(average_scores[3]),
'{{SST_RUBRIC}}': getRubric(average_scores[4]),
'{{ART_RUBRIC}}': getRubric(average_scores[5]),
'{{PRETECH_RUBRIC}}': getRubric(average_scores[6]),
'{{AGRIC_RUBRIC}}': getRubric(average_scores[7]),
'{{CRE_RUBRIC}}': getRubric(average_scores[8]),

'{{TOTAL_CAT1}}': total1.toString(),
'{{TOTAL_CAT2}}': total2.toString(),
'{{TOTAL_AVG}}': total3.toString(),

'{{MEAN_CAT1}}': mean1,
'{{MEAN_CAT2}}': mean2,
'{{MEAN_AVG}}': mean3,

'{{MEAN_CAT1_RUBRIC}}': rubric1,
'{{MEAN_CAT2_RUBRIC}}': rubric2,
'{{MEAN_AVG_RUBRIC}}': rubric3,
'{{AVG_RUBRIC}}': rubric3,
'{{AVG_COMMENT}}': getGeneralComment(rubric3),

'{{TERM}}': term,
'{{YEAR}}': year,
'{{GRADE}}': grade
};
for (let key in placeholders) {
body.replaceText(key, placeholders[key]);
}

const logoTag = body.findText('{{SCHOOL_LOGO}}');
if (logoTag) {
const element = logoTag.getElement().getParent();
const index = body.getChildIndex(element);
```



```
const headerTable = body.insertTable(0);
const row = headerTable.appendTableRow();
const logoCell = row.appendTableCell();
const textCell = row.appendTableCell();

const logo = logoCell.insertImage(0, logoBlob);
logo.setWidth(110).setHeight(120);
logoCell.setWidth(100);
textCell.setWidth(400);

textCell.setText(`🏫 Mungakha Junior School\n📍 Bungoma, Kenya\n📘 ${grade}, ${term} Year, ${year}`);
textCell.setFontSize(20).setBold(true);

body.removeChild(element);
}

doc.saveAndClose();

insertStudentChart(name, cat1_scores, cat2_scores, folder, docCopy.getId());

docCopy.setSharing(DriveApp.Access.ANYONE_WITH_LINK, DriveApp.Permission.VIEW);
Utilities.sleep(1000);
const downloadLink = docCopy.getUrl();

const reopenedDoc = DocumentApp.openById(docCopy.getId());
const reopenedBody = reopenedDoc.getBody();
reopenedBody.appendParagraph('\nDownload Your Report Card:');
reopenedBody.appendParagraph(downloadLink).setLinkUrl(downloadLink);
reopenedDoc.saveAndClose();

const pdf = docCopy.getAs(MimeType.PDF);
const pdfFile = folder.createFile(pdf);

if (email) {
GmailApp.sendEmail(email, 'Your Report Card',
```



```
`Dear ${name},\n\nAttached is the report card for ${term} ${year}.\n\nDownload Link:\n${downloadLink}\n\nBest regards,\nMungakha Junior School`, {
attachments: [pdfFile],
name: 'School Reports'
});
}

} catch (error) {
Logger.log(`⚠ Error processing student at row ${i + 1}: ${error}`);
}

}

SpreadsheetApp.flush();

generateRubricSummary(data, folder);

SpreadsheetApp.getUi().alert("✅ Report Cards with Charts and Download Links Generated Successfully!");
}




function getRubric(score) {
score = Number(score);
if (score >= 90) return "E.E ~AL8";
if (score >= 75) return "E.E ~AL7";
if (score >= 60) return "M.E ~AL6";
if (score >= 50) return "M.E ~AL5";
if (score >= 35) return "A.E ~AL4";
if (score >= 25) return "A.E ~AL3";
if (score >= 15) return "B.E ~AL2";
return "B.E ~AL1";
}
```



```
// Sidebar + Menu
function onOpen() {
SpreadsheetApp.getUi()
.createMenu('📄 Report Cards')
.addItem('📎 Open Sidebar', 'showSidebar')
.addToUi();
}

function showSidebar() {
const html = HtmlService.createHtmlOutputFromFile('sidebar')
.setTitle('Report Card Generator')
.setWidth(300);
SpreadsheetApp.getUi().showSidebar(html);
SpreadsheetApp.getActiveSpreadsheet().toast('Sidebar opened successfully!');
}
function getGeneralComment(rubric) {
switch (rubric) {
case "E.E ~AL8":
return "Exceptionally excellent performance! Hongera! Keep up the great work and continue aiming high.";
case "E.E ~AL7":
return "Excellent performance! Hongera! Keep up the great work and continue aiming high.";
case "M.E ~AL6":
return " Exceptionally good job! Vyema! You are meeting the required standards. Strive for even greater heights.";
case "M.E ~AL5":
return "Good job! Vyema! You are meeting the required standards. Strive for even greater heights.";
case "A.E ~AL4":
return "Exceptionally fair performance. Jikakamue! More effort and focus will lead to improvement.";
case "A.E ~AL3":
return "Fair performance. Jikakamue! More effort and focus will lead to improvement.";
case "B.E ~AL2":
return "Needs improvement. Jitahidi! Let's work together to achieve better results.";
case "B.E ~AL1":
return "Serously needs improvement. Jitahidi! Let's work together to achieve better results.";
default:
```

```
return "No comment available.";
}
}
function generateRubricSummary(data, folder) {
let rubricCount = {
'E.E ~AL8': 0,
'E.E ~AL7': 0,
'M.E ~AL6': 0,
'M.E ~AL5': 0,
'A.E ~AL4': 0,
'A.E ~AL3': 0,
'B.E ~AL2': 0,
'B.E ~AL1': 0
}

for (let i = 1; i < data.length; i++) {
const cat1 = Number(data[i][2]); // Example column index for ENG_CAT1
const cat2 = Number(data[i][3]); // Example column index for ENG_CAT2
const avg = ((cat1 + cat2) / 2).toFixed(2);

const rubric = getRubric(avg);
if (rubric in rubricCount) {
rubricCount[rubric]++;
}
}

const doc = DocumentApp.create("Rubric Summary");
const body = doc.getBody();

body.appendParagraph("Rubric Summary Report");
for (let key in rubricCount) {
body.appendParagraph(`${key}: ${rubricCount[key]}`);
}

const file = DriveApp.getFileById(doc.getId());
```

```
folder.addFile(file);
DriveApp.getRootFolder().removeFile(file); // Remove from root if needed
}



function insertStudentChart(name, cat1_scores, cat2_scores, folder, docId) {
const ss = SpreadsheetApp.getActiveSpreadsheet();
// Check if a temp chart sheet already exists and delete
const existing = ss.getSheetByName('TempChart');
if (existing) ss.deleteSheet(existing);
const chartSheet = ss.insertSheet('TempChart');
chartSheet.getRange(1, 1).setValue("Assessment");
chartSheet.getRange(1, 2).setValue("CAT 1");
chartSheet.getRange(1, 3).setValue("CAT 2");

const subjects = ["ENG", "KIS", "MATH", "SCI", "SST", "ART", "PRETECH", "AGRIC", "CRE"];
for (let i = 0; i < subjects.length; i++) {
chartSheet.getRange(i + 2, 1).setValue(subjects[i]);
chartSheet.getRange(i + 2, 2).setValue(cat1_scores[i]);
chartSheet.getRange(i + 2, 3).setValue(cat2_scores[i]);
}

const chart = chartSheet.newChart()
.setChartType(Charts.ChartType.COLUMN)
.addRange(chartSheet.getRange(1, 1, subjects.length + 1, 3))
.setPosition(1, 5, 0, 0)
.build();

chartSheet.insertChart(chart);

// Export the chart as image
const blob = chartSheet.getCharts()[0].getAs('image/png');
const imgFile = folder.createFile(blob).setName(`${name}_Chart.png`);
```

```
const doc = DocumentApp.openById(docId);
const body = doc.getBody();
body.appendParagraph("\nPerformance Chart:");
body.appendImage(blob);
doc.saveAndClose();

// Delete chart sheet after use
ss.deleteSheet(chartSheet);
}
```

---

sidebar.html

```
<!-- Sidebar.html -->
<!DOCTYPE html>
<html>
<head>
<base target="_top">
<style>
body {
font-family: "Roboto", sans-serif;
padding: 10px;
}
button {
background-color: #1a73e8;
color: white;
border: none;
```

```
padding: 10px 20px;
font-size: 14px;
border-radius: 5px;
cursor: pointer;
}
button:hover {
background-color: #155ab6;
}
</style>
</head>
<body>
<h3>Generate Report Cards</h3>
<p>This will create report cards for all learners, with charts and email delivery.</p>
<button onclick="generate()">Generate Now</button>
<div id="status" style="margin-top: 10px;"></div>
<script>
function generate() {
document.getElementById("status").innerText = "Processing...";
google.script.run
.withSuccessHandler(function() {
document.getElementById("status").innerText = "Report Cards Generated Successfully!";
})
.withFailureHandler(function(error) {
document.getElementById("status").innerText = "Error: " + error.message;
})
.generateReportCardsWithCharts();
}
</script>
</body>
</html>
```

---

template

{{SCHOOL_LOGO}}

Student Report Card for Grade 8

Student Name: {{NAME}}

Admission Number: {{ADM}}

Subjects and Scores:

Learning Areas	Cat1 Scores	Cat1 Rubrics	Cat2 Scores	Cat2 Rubrics	Average Scores	Rubric Levels
English	{{ENG_CAT1}}	{{ENG_CAT1_RUBRIC}}	{{ENG_CAT2}}	{{ENG_CAT2_RUBRIC}}	{{ENG}}	{{ENG_RUBRIC}}
Kiswahili	{{KIS_CAT1}}	{{KIS_CAT1_RUBRIC}}	{{KIS_CAT2}}	{{KIS_CAT2_RUBRIC}}	{{KIS}}	{{KIS_RUBRIC}}
Maths	{{MATH_CAT1}}	{{MATH_CAT1_RUBRIC}}	{{MATH_CAT2}}	{{MATH_CAT2_RUBRIC}}	{{MATH}}	{{MATH_RUBRIC}}
Int Scie	{{SCI_CAT1}}	{{SCI_CAT1_RUBRIC}}	{{SCI_CAT2}}	{{SCI_CAT2_RUBRIC}}	{{SCI}}	{{SCI_RUBRIC}}
SST	{{SST_CAT1}}	{{SST_CAT1_RUBRIC}}	{{SST_CAT2}}	{{SST_CAT2_RUBRIC}}	{{SST}}	{{SST_RUBRIC}}

C.Arts & S	{{ART_CAT1}}	{{ART_CAT1_RUBRIC}}	{{ART_CAT2}}	{{ART_CAT2_RUBRIC}}	{{ART}}	{{ART_RUBRIC}}
Pretech	{{PRETECH_CAT1}}	{{PRETECH_CAT1_RUBRIC}}	{{PRETECH_CAT2}}	{{PRETECH_CAT2_RUBRIC}}	{{PRETECH}}	{{PRETECH_RUBRIC}}
Agrics	{{AGRIC_CAT1}}	{{AGRIC_CAT1_RUBRIC}}	{{AGRIC_CAT2}}	{{AGRIC_CAT2_RUBRIC}}	{{AGRIC}}	{{AGRIC_RUBRIC}}
C.R.E	{{CRE_CAT1}}	{{CRE_CAT1_RUBRIC}}	{{CRE_CAT2}}	{{CRE_CAT2_RUBRIC}}	{{CRE}}	{{CRE_RUBRIC}}

Totals and Averages:

Category	Total	Mean	Rubric Level
CAT 1 SCORES	{{TOTAL_CAT1}}	{{MEAN_CAT1}}	{{MEAN_CAT1_RUBRIC}}
CAT 2 SCORES	{{TOTAL_CAT2}}	{{MEAN_CAT2}}	{{MEAN_CAT2_RUBRIC}}
AVERAGE SCORES	{{TOTAL_AVG}}	{{MEAN_AVG}}	{{MEAN_AVG_RUBRIC}}

Class Teacher's Comment: ***{{AVG_COMMENT}}***

H.O.I'S Comment:

……..

H.O.I 's Signature: _____ Date: ……

Closing Date: 4th of April, 2025

Opening Date:28th of April,2025