Use this library for creating PDF:  
<https://www.npmjs.com/package/html-pdf/v/3.0.1>

Code from Result Pricing

const fs = require('fs')

const pdf = require('html-pdf')

// eslint-disable-next-line import/no-extraneous-dependencies

const request = require('request')

const options = {

format: 'A4',

phantomPath: '/usr/local/bin/phantomjs'

}

const downloadToBase64 = async (req, res) => {

const {

companyLogo

} = req.user

if (!companyLogo) {

res.status(400).send({

message: 'Company Logo not Found'

})

}

const download = function (uri, filename, callback) {

try {

request.head(uri, (res) => {

request(uri).pipe(fs.createWriteStream(filename)).on('close', callback)

})

} catch (error) {

res.status(400).send({

message: 'Invalid Image URL',

error: error

})

}

}

download(companyLogo, 'google.png', () => {

const fileBuffer = fs.readFileSync('google.png')

const contentsInBase64 = fileBuffer.toString('base64')

excelToHtml(contentsInBase64, req, res)

})

}

const excelToHtml = async (imgData, req, res) => {

console.log('USER', req.user)

const {

customField,

homeOwner,

categories,

proposalName,

poolPerimeter,

poolSqft,

depth1,

depth2,

depth3,

spaPerimeter,

spaRaisedAbovePool,

spaSqft,

// spaDepth,

totalRetail,

notes,

} = req.body

const User = req.user

const createdAt = req.body.created\_at.split('T')[0]

const ownerName = `${homeOwner[0].persons[0].firstName} ${homeOwner[0].persons[0].lastName}`

const depth = ((parseFloat(depth1) + parseFloat(depth2) + parseFloat(depth3)) / 3).toFixed(2)

const gallons = ((poolSqft) \* (depth) \* (7.5)).toFixed(2)

const CategoryDataHTML = getCategoryData(categories)

const CustomFieldDataHTML = getCustomFieldData(customField)

const htmlFile = `<!DOCTYPE html>

<html

lang="en"

xmlns="http://www.w3.org/1999/xhtml"

xmlns:v="urn:schemas-microsoft-com:vml"

xmlns:o="urn:schemas-microsoft-com:office:office"

>

<head>

<meta charset="utf-8" />

<!-- utf-8 works for most cases -->

<meta name="viewport" content="width=device-width" />

<!-- Forcing initial-scale shouldn't be necessary -->

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<!-- Use the latest (edge) version of IE rendering engine -->

<meta name="x-apple-disable-message-reformatting" />

<!-- Disable auto-scale in iOS 10 Mail entirely -->

<title></title>

<!-- The title tag shows in email notifications, like Android 4.4. -->

<link

href="https://fonts.googleapis.com/css?family=Lato:300,400,700"

rel="stylesheet"

/>

<!-- CSS Reset : BEGIN -->

<style>

/\* What it does: Remove spaces around the email design added by some email clients. \*/

/\* Beware: It can remove the padding / margin and add a background color to the compose a reply window. \*/

html,

body {

margin: 0 auto !important;

padding: 0 !important;

height: 100% !important;

width: 100% !important;

background: #f1f1f1;

}

/\* What it does: Stops email clients resizing small text. \*/

\* {

-ms-text-size-adjust: 100%;

-webkit-text-size-adjust: 100%;

}

/\* What it does: Centers email on Android 4.4 \*/

div[style\*="margin: 16px 0"] {

margin: 0 !important;

}

/\* What it does: Stops Outlook from adding extra spacing to tables. \*/

table,

td {

font-size:14px;

mso-table-lspace: 0pt !important;

mso-table-rspace: 0pt !important;

}

/\* What it does: Fixes webkit padding issue. \*/

table {

border-spacing: 0 !important;

border-collapse: collapse !important;

table-layout: fixed !important;

margin: 0 40px;

}

th {

text-align: left;

}

tbody {

}

/\* What it does: Uses a better rendering method when resizing images in IE. \*/

img {

-ms-interpolation-mode: bicubic;

}

/\* What it does: Prevents Windows 10 Mail from underlining links despite inline CSS. Styles for underlined links should be inline. \*/

a {

text-decoration: none;

}

/\* What it does: A work-around for email clients meddling in triggered links. \*/

\*[x-apple-data-detectors], /\* iOS \*/

.unstyle-auto-detected-links \*,

.aBn {

border-bottom: 0 !important;

cursor: default !important;

color: inherit !important;

text-decoration: none !important;

font-size: inherit !important;<tr>

font-family: inherit !important;

font-weight: inherit !important;

line-height: inherit !important;

}

/\* What it does: Prevents Gmail from displaying a download button on large, non-linked images. \*/

.a6S {

display: none !important;

opacity: 0.01 !important;

}

/\* What it does: Prevents Gmail from changing the text color in conversation threads. \*/

.im {

color: inherit !important;

}

/\* If the above doesn't work, add a .g-img class to any image in question. \*/

img.g-img + div {

display: none !important;

}

/\* What it does: Removes right gutter in Gmail iOS app: https://github.com/TedGoas/Cerberus/issues/89 \*/

/\* Create one of these media queries for each additional viewport size you'd like to fix \*/

/\* iPhone 4, 4S, 5, 5S, 5C, and 5SE \*/

@media only screen and (min-device-width: 320px) and (max-device-width: 374px) {

u ~ div .email-container {

min-width: 320px !important;

}

}

/\* iPhone 6, 6S, 7, 8, and X \*/

@media only screen and (min-device-width: 375px) and (max-device-width: 413px) {

u ~ div .email-container {

min-width: 375px !important;

}

}

/\* iPhone 6+, 7+, and 8+ \*/

@media only screen and (min-device-width: 414px) {

u ~ div .email-container {

min-width: 414px !important;

}

}

</style>

<!-- CSS Reset : END -->

<!-- Progressive Enhancements : BEGIN -->

<style>

.primary {

background: #30e3ca;

}

.bg\_white {

background: #ffffff;

}

.bg\_light {

background: #fafafa;

}

.bg\_black {

background: #000000;

}

.bg\_dark {

background: rgba(0, 0, 0, 0.8);

}

.email-section {

padding: 2.5em;

}

.border-bottom {

border-bottom: 1px solid rgba(128, 128, 128, 0.37);

}

/\*BUTTON\*/

.btn {

padding: 10px 15px;

display: inline-block;

}

.btn.btn-primary {

border-radius: 0px;

background: #252525;

color: #ffffff;

}

.btn.btn-white {

border-radius: 5px;

background: #ffffff;

color: #000000;

}

.btn.btn-white-outline {

border-radius: 5px;

background: transparent;

border: 1px solid #fff;

color: #fff;

}

.btn.btn-black-outline {

border-radius: 0px;

background: transparent;

border: 2px solid #000;

color: #000;

font-weight: 700;

}

h1,

h2,

h3,

h4,

h5,

h6 {

font-family: "Lato", sans-serif;

color: #000000;

margin-top: 0;

font-weight: 400;

}

body {

font-family: "Lato", sans-serif;

font-weight: 400;

font-size: 15px;

line-height: 1.8;

color: rgba(0, 0, 0, 0.534);

}

a {

color: #30e3ca;

}

table {

}

/\*LOGO\*/

.logo h1 {

margin: 0;

}

.logo h1 a {

color: #30e3ca;

font-size: 24px;

font-weight: 700;

font-family: "Lato", sans-serif;

}

/\*HERO\*/

.hero {

position: relative;

z-index: 0;

}

.hero .text {

color: rgba(0, 0, 0, 0.3);

}

.hero .text h2 {

color: #000;

font-size: 40px;

margin-bottom: 0;

font-weight: 400;

line-height: 1.4;

}

.hero .text h3 {

font-size: 24px;

font-weight: 300;

}

.hero .text h2 span {

font-weight: 600;

color: #30e3ca;

}

/\*HEADING SECTION\*/

.heading-section {

}

.heading-section h2 {

color: #000000;

font-size: 28px;

margin-top: 0;

line-height: 1.4;

font-weight: 400;

}

.heading-section .subheading {

margin-bottom: 20px !important;

display: inline-block;

font-size: 13px;

text-transform: uppercase;

letter-spacing: 2px;

color: rgba(0, 0, 0, 0.4);

position: relative;

}

.heading-section .subheading::after {

position: absolute;

left: 0;

right: 0;

bottom: -10px;

content: "";

width: 100%;

height: 2px;

background: #30e3ca;

margin: 0 auto;

}

.heading-section-white {

color: rgba(255, 255, 255, 0.8);

}

.heading-section-white h2 {

line-height: 1;

padding-bottom: 0;

}

.heading-section-white h2 {

color: #ffffff;

}

.heading-section-white .subheading {

margin-bottom: 0;

display: inline-block;

font-size: 13px;

text-transform: uppercase;

letter-spacing: 2px;

color: rgba(255, 255, 255, 0.4);

}

ul.social {

padding: 0;

}

ul.social li {

display: inline-block;

margin-right: 10px;

}

.notes{

width: 500px;

margin: 0 auto;

height: 100px;

border: 1px solid grey;

border-radius: 10px;

padding: 15px;

}

/\*FOOTER\*/

.footer {

border-top: 1px solid rgba(0, 0, 0, 0.05);

color: rgba(0, 0, 0, 0.5);

}

.footer .heading {

color: #000;

font-size: 20px;

}

.footer ul {

margin: 0;

padding: 0;

}

.footer ul li {

list-style: none;

margin-bottom: 10px;

}

.footer ul li a {

color: rgba(0, 0, 0, 1);

}

.company-logo{

max-width: 200px;

}

@media screen and (max-width: 500px) {

}

</style>

</head>

<body

width="100%"

style="

margin: 0;

padding: 0 !important;

mso-line-height-rule: exactly;

background-color: #f1f1f1;

"

>

<div id="propsal" style="width: 100%; background-color: #f1f1f1">

<div

style="

display: none;

font-size: 1px;

max-height: 0px;

max-width: 0px;

opacity: 0;

overflow: hidden;

mso-hide: all;

font-family: sans-serif;

"

>

&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;&zwnj;&nbsp;

</div>

<div

style="

max-width: 600px;

margin: 0 auto;

background-color: #fff;

padding-bottom: 50px;

"

class="email-container"

>

<!-- BEGIN BODY -->

<table

align="center"

role="presentation"

cellspacing="0"

cellpadding="0"

border="0"

width="100%"

style="margin: 30px auto"

class="bg\_white border-bottom"

>

<tr>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

></td>

</tr>

<!-- end tr -->

<tr>

<td

valign="middle"

class="hero bg\_white"

style="padding: 3em 0 2em 0; background-color: #2b308e"

>

<h1

class="heading-section-white"

style="text-align: center; margin: 10px auto"

>

<img class="company-logo"

src="data:image/png;base64,${imgData}"

alt="Company Logo"

/>

</h1>

<h5

class="heading-section-white"

style="text-align: center; margin: 10px auto"

>

${User.phone}

</h5>

<h5

class="heading-section-white"

style="text-align: center; margin: 10px auto"

>

${User.address.subDivision} ${User.address.street}, ${User.address.city}

</h5>

</td>

</tr>

<!-- end tr -->

<tr></tr>

<!-- end tr -->

<!-- 1 Column Text + Button : END -->

</table>

<table

align="center"

role="presentation"

cellspacing="0"

cellpadding="0"

border="0"

width="100%"

style="margin: 30px auto"

class="border-bottom"

>

<tr>

<td valign="middle" class="hero bg\_white">

<table>

<thead>

<tr>

<th>HomeOwner Info</th>

</tr>

</thead>

<tbody>

<tr>

<td>Name</td>

<td>${ownerName}</td>

</tr>

<tr>

<td>Address</td>

<td>${homeOwner[0].address.subDivision} ${homeOwner[0].address.street}, ${homeOwner[0].address.city}</td>

</tr>

<tr>

<td>Phone</td>

<td>${homeOwner[0].persons[0].phone}</td>

</tr>

</tbody>

</table>

</td>

<td valign="middle" class="hero bg\_white">

<table>

<thead>

<tr>

<th>Proposal Details</th>

</tr>

</thead>

<tbody>

<tr>

<td>Name</td>

<td>${proposalName}</td>

</tr>

<tr>

<td>Date</td>

<td>${createdAt}</td>

</tr>

<tr>

<td>Designer</td>

<td> ${User.firstName} ${User.lastName}</td>

</tr>

</tbody>

</table>

</td>

</tr>

</table>

<table

align="center"

role="presentation"

cellspacing="0"

cellpadding="0"

border="0"

width="100%"

style="margin: 30px auto"

class="bg\_white "

>

<tr>

<thead>

<th style="padding: 0em 0 0 2.5em" colspan="8">General Specifications</th>

</thead>

</tr>

<tr>

<td

colspan="3"

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Pool Perimeter

</td>

<td valign="top" class="bg\_white" >

${poolPerimeter}

</td>

<td

colspan="3"

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Spa Perimeter

</td>

<td valign="top" class="bg\_white">

${spaPerimeter}</td>

</tr>

<!-- end tr -->

<tr>

<td

colspan="3"

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Pool Square footage

</td>

<td valign="top" class="bg\_white" >

${poolSqft}</td>

<td

colspan="3"

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Spa Square footage

</td>

<td valign="top" class="bg\_white" >

${spaSqft}</td>

</tr>

<!-- end tr -->

<tr>

<td

colspan="3"

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Depth Profile

</td>

<td valign="top" class="bg\_white" >

${depth}</td>

<td

colspan="3"

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Spa height

</td>

<td valign="top" class="bg\_white">

${spaRaisedAbovePool}</td>

</tr>

<tr class="border-bottom" colspan='8'>

<td valign="top" class="bg\_white" style="padding: 0em 2.5em 0 2.5em;" colspan="3">

Gallons

</td>

<td valign="top" class="bg\_white" colspan='5'>

${gallons}

</td>

</tr>

<!-- end tr -->

<!-- 1 Column Text + Button : END -->

</table>

${CategoryDataHTML}

${CustomFieldDataHTML}

<table

align="center"

role="presentation"

cellspacing="0"

cellpadding="0"

border="0"

width="100%"

style="margin: 30px auto"

class="bg\_white"

>

<tr>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Retail Price

</td>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

$ ${totalRetail}

</td>

</tr>

<!-- end tr -->

<tr>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Discount

</td>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

$0

</td>

</tr>

<!-- end tr -->

<tr>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Subtotal

</td>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

$0

</td>

</tr>

<!-- end tr -->

<tr>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Taxes:

</td>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

$0

</td>

</tr>

<tr>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

Total Value:

</td>

<td

valign="top"

class="bg\_white"

style="padding: 0em 2.5em 0 2.5em"

>

$0

</td>

</tr>

<!-- 1 Column Text + Button : END -->

</table>

<table align="center" role="presentation" cellspacing="0" cellpadding="0" border="0" width="100%" style="margin:30px auto;" class="bg\_white">

<tr>

<td class="bg\_white">

<div class="notes">

Notes: ${notes}

</div>

</td>

</tr>

<!-- 1 Column Text + Button : END -->

</table>

</div>

</div>

</body>

</html>`

pdf.create(htmlFile, options).toFile(`pdfs/${new Date().getTime()}.pdf`, (err, result) => {

if (err) {

console.log(err)

return res

.status(400)

.json({

message: err.message

})

}

const newPath = result.filename.split('/')

const filePath = `pdfs/${newPath[newPath.length - 1]}`

res.download(filePath, ownerName, (err) => {

if (err) {

console.log(err)

res.status(400).send({

message: 'Error in Sending the pproposal'

})

}

fs.unlinkSync('./google.png')

fs.unlinkSync(filePath)

})

})

}

const getCategoryData = (categories) => {

const CategoryData = []

categories.forEach(category => {

let total = 0

const itemsList = []

category.sub\_Categories.forEach((subCategory) => {

if (subCategory.sub\_sub\_Categories.length > 0) {

subCategory.sub\_sub\_Categories.forEach((subSubCat) => {

total += subSubCat.quantity \* subSubCat.sub\_sub\_category.price

itemsList.push({

itemName: subSubCat.sub\_sub\_category.name,

itemQuantity: subSubCat.quantity

})

})

} else {

total += subCategory.quantity \* subCategory.sub\_category.price

itemsList.push({

itemName: subCategory.sub\_category.name,

itemQuantity: subCategory.quantity

})

}

})

CategoryData.push({

categoryName: category.category.name,

itemsList,

total

})

})

return getHtmlData(CategoryData)

}

const getCustomFieldData = (customFields) => {

const itemsList = []

let total = 0

customFields.forEach(field => {

total += field.retailPrice

itemsList.push({

itemName: field.name,

itemQuantity: 1

})

})

return getHtmlData([{

categoryName: 'Custom Field/item',

itemsList,

total

}])

}

const getHtmlData = (dataArray) => {

let categoryHtml = ''

dataArray.forEach(category => {

let catHtml = `<table align="center" role="presentation" cellspacing="0" cellpadding="0" border="0" width="100%"

style="margin: 30px auto" class="bg\_white border-bottom">

<tr>

<thead>

<th style="padding: 0em 0 0 2.5em">${category.categoryName}</th>

<th style="padding: 0em 0 0 2.5em">Quantity</th>

</thead>

</tr>`

category.itemsList.forEach(item => {

const itemHtml = `<tr>

<td valign="top" class="bg\_white" style="padding: 0em 2.5em 0 2.5em">

${item.itemName}

</td>

<td valign="top" class="bg\_white" style="padding: 0em 2.5em 0 2.5em">

${item.itemQuantity}

</td>

</tr>`

catHtml += itemHtml

})

catHtml += `<tr>

<td valign="top" class="bg\_white" style="padding: 0em 2.5em 0 2.5em"></td>

<td valign="top" class="bg\_white" style="padding: 0em 2.5em 0 2.5em">

Total Value:${category.total}

</td>

</tr>

<!-- 1 Column Text + Button : END -->

</table>`

categoryHtml += catHtml

})

return categoryHtml

}

module.exports = {

downloadToBase64

}