Generate PDF FILES FROM DOTNET BACKEND

1. Add this package

dotnet add package Polybioz.HtmlRenderer.PdfSharp.Core --version 1.0.0

2.Add the following NameSpace

using PdfSharpCore;
using PdfSharpCore.Pdf;
using TheArtOfDev.HtmlRenderer.PdfSharp;

3. Sample Backend Should Look Like This

```
[HttpGet("generator")]
public async Task<IActionResult>GenPdfWithImage(string Id){

var student = context.Students.FirstOrDefault(r=>r.StudentId==Id);
var inst = context.Instituitions.FirstOrDefault(r=>r.Id>0);
var auth = context.AuthenticationModels.FirstOrDefault(r=>r.UserId==Id);
if(student==null || inst==null || auth==null){
return BadRequest("Student Not Found");
}
var adfee = context.AdmissionFees.FirstOrDefault(r => r.Level == student.Level
&& r.AcademicYear == student.TheAcademicYear);
var otherfee = context.Fees.Where(r => r.Level == student.Level &&
r.AcademicYear == student.TheAcademicYear && r.AcademicTerm ==
student.TheAcademicTerm).Sum(r => r.Amount);

double? fees;

if (adfee != null)
{
fees = adfee.Amount + otherfee;
}
else
{
fees = otherfee;
}




var document = new PdfDocument();

string studentImage = "http://" + HttpContext.Request.Host.Value + "/" +
student.ProfilePic;
string schoolImage = "http://" + HttpContext.Request.Host.Value + "/"+
inst.Logo;
var schoolname = inst.SchoolName;
var Location = inst.Location;
var Registerer = inst.AdminName;
var fullname = student.FirstName+" "+ student.OtherName+ " "+student.LastName;


string htmlelement = $@"<!DOCTYPE html>
<html>
<head>
<style>
body {{
font-family: 'Open Sans', sans-serif;
font-size: 14px;
background-color: #f4f4f4;
margin: 0;
padding: 0;
}}

.container {{
max-width: 600px;
margin: 0 auto;
background-color: #ffffff;
padding: 20px;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
animation: fade-in 0.5s ease-out;
}}

.container img {{
width: 120px;
height: auto;
display: block;
margin: 0 auto;
}}

.header {{
font-size: 1.6rem;
font-weight: bold;
margin-top: 20px;
}}

.center{{
text-align: center;
}}

.locbold {{
font-size: 15px;
float: left;
text-align: center;
}}
```

```css
.school-name {{
font-size: 1.2rem;
font-weight: bold;
margin-top: 20px;
}}

.content {{
text-align: justify;
}}

.contentp {{
font-size: 18px;
margin-bottom: 20px;
line-height: 1.5;
font-weight: 900;
}}


.spacer{{
margin-left: 200px;
}}
.bold{{
font-size: 15px;
float: left; /* Float the left div to the left */
text-align: left; /* Align text to the left within the left div */

}}
.lightbold{{
font-size: 15px;
}}
.school{{
font-size: 30px;
}}

/* Add individual styles for .left, .center, and .right */
.left {{
float: left; /* Float the left div to the left */
width: 30%; /* Adjust the width as needed */
padding: 10px; /* Add padding for spacing */
box-sizing: border-box; /* Include padding and border in the width */
text-align: left; /* Align text to the left within the left div */
}}



.right {{
float: right; /* Float the right div to the left */
padding: 10px; /* Add padding for spacing */
box-sizing: border-box; /* Include padding and border in the width */
text-align: right; /* Align text to the right within the right div */
margin-left:120px;
margin-top: -2300rem;

}}




</style>
</head>
<body>
<div class='container'>
<div class='center'>
<img src='{schoolImage}' alt='School Logo' />
</div>
<br/>
<div >
<span class='bold'>Dear {fullname},</span>
<span class='spacer lightbold'>{student.AdmissionDate}</span>
</div>

<br/>
<div class='center'>
<div class='school'>{schoolname}</div>
<div class='locbold'>{Location}</div>

</div>
<hr/>
<div class='content'>
<p class='contentp'>1. We are delighted to extend an offer of admission to
<b>{schoolname}</b> for this academic year. It brings us great pleasure to
welcome you to our school community, and we are excited about the prospect of
having you as a student in our school.</p>
<p class='contentp'>2. Your application to {schoolname} was thoroughly reviewed
by our admissions committee, and we were impressed by your academic potential,
your eagerness to learn, and the positive qualities you exhibited during the
admission process. Your enthusiasm for education and your readiness to engage
with our school curriculum make you an excellent fit for our school.</p>
<p class='contentp'>3. Your student identification number is
{student.StudentId}. This is the number you will use throughout your studies at
our school. Your password is {auth.RawPassword}; please be sure not to misplace
this information.</p>
<p class='contentp'>4. For the {student.TheAcademicYear} academic year, the
Admission Fee is {adfee?.Amount}, and the Tuition Fee, along with other fees,
for {student.Level} {student.TheAcademicTerm} is {otherfee}.</p>
<p class='contentp'>5. Your total provisional fees for {student.Level}
{student.TheAcademicTerm} in the {student.TheAcademicYear} academic year amount
to {fees}. Please make this payment to the school's accountant only.</p>
<p class='contentp'>6. If you have any questions or require further assistance,
please feel free to contact our admissions office. We look forward to having you
```

```
join our {schoolname} family and embark on an exciting educational adventure. We are confident that your time with us will be filled with learning, growth, and lasting friendships.</p>
<p class='contentp'>7. Congratulations once again on your admission to {schoolname}. We can't wait to get to know you better and watch you thrive as a member of our school community.</p>
<div class='right'>
Yours Faithfully,<br/>

.........................<br/>
{inst.AdminName}<br/>
{inst.Position}
</div>
</div>
<br/><br/>

<div class='center'>
<img src='{studentImage}' alt='Student Image' />
</div>

</div>
</body>
</html>";



PdfGenerator.AddPdfPages(document,htmlelement,PageSize.A4);
byte[] response = null;
using (MemoryStream ms = new MemoryStream()){
document.Save(ms);
response = ms.ToArray();
}
return File(response,"application/pdf",$"{fullname}.pdf");
}
```

4. To Auto Download, The Sample Frontend Function Should Be Like This

```
const studentDetails = async (event) => {
event.preventDefault();

Show.showLoading("Processing Data");

const URL = `api/Accounting/PayFees?StudentId=${studentId}&StaffId=${userInfo.staffID}`
try {
const response = await fetch(apiServer + URL, {
method: "POST",
headers: {
"Content-Type": "application/json", // Set the Content-Type header
},
body: JSON.stringify({ amount,paymentMethod,action }),
});
const data = await response.json();
if (response.ok) {
handleGeneratePDF(data.transactionId)
} else {
Show.Attention("All fields are required");
}
} catch (err) {
Show.Attention("An error has occurred");
}
};

const handleGeneratePDF = async (PayId) => {
try {
const URL = `api/ThePDFS/FeesPayment?Id=${theStudent.studentId}&PayId=${PayId}`
// Send a request to the backend to generate the PDF
const response = await fetch(apiServer + URL, {
method: 'GET',
});
if (response.ok) {
// Convert the response to a blob
const blob = await response.blob();
// Create a URL for the blob
const url = window.URL.createObjectURL(blob);
// Create an anchor element to trigger the download
const a = document.createElement('a');
a.href = url;
a.download = `${theStudent?.firstName}_${theStudent?.lastName}.pdf`
document.body.appendChild(a);
a.click();
// Clean up the URL and remove the anchor element
window.URL.revokeObjectURL(url);
document.body.removeChild(a);



} else {
// Handle HTTP error responses
console.error('HTTP Error:', response.status, response.statusText);
}
} catch (error) {
// Handle other errors (e.g., network issues)
console.error('Error generating PDF:', error);
} finally {
Show.hideLoading();
Show.Success("Payment made successfully");
window.location.reload();
}
};
```