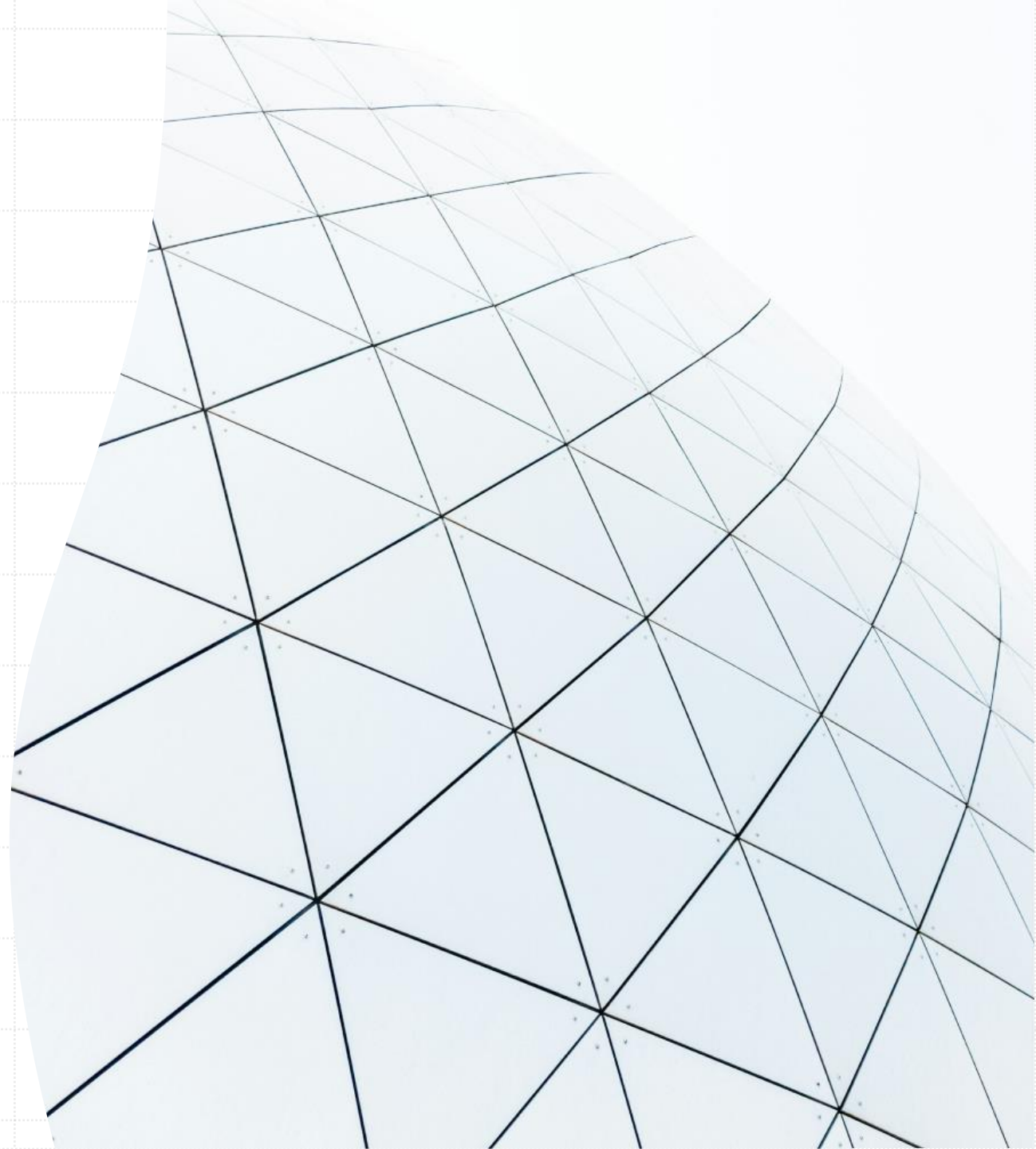


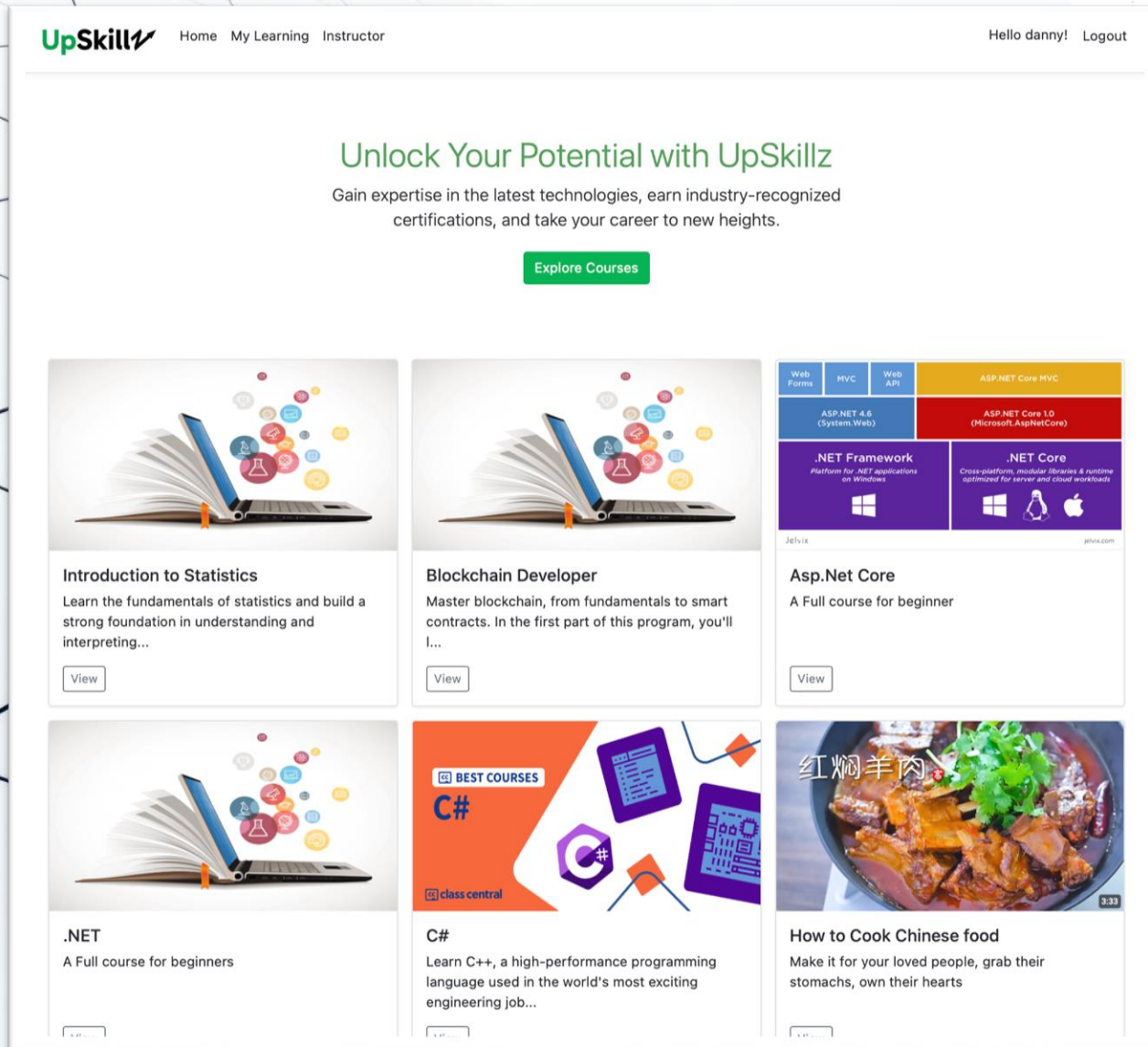


Project presentation

Application development

Tamara Plante | Iana Setrakova | Dong Huang





Project purpose

Online learning platform

Enables students to:

- Enroll in courses and access multimedia lessons.
- Track progress and mark lessons as completed.

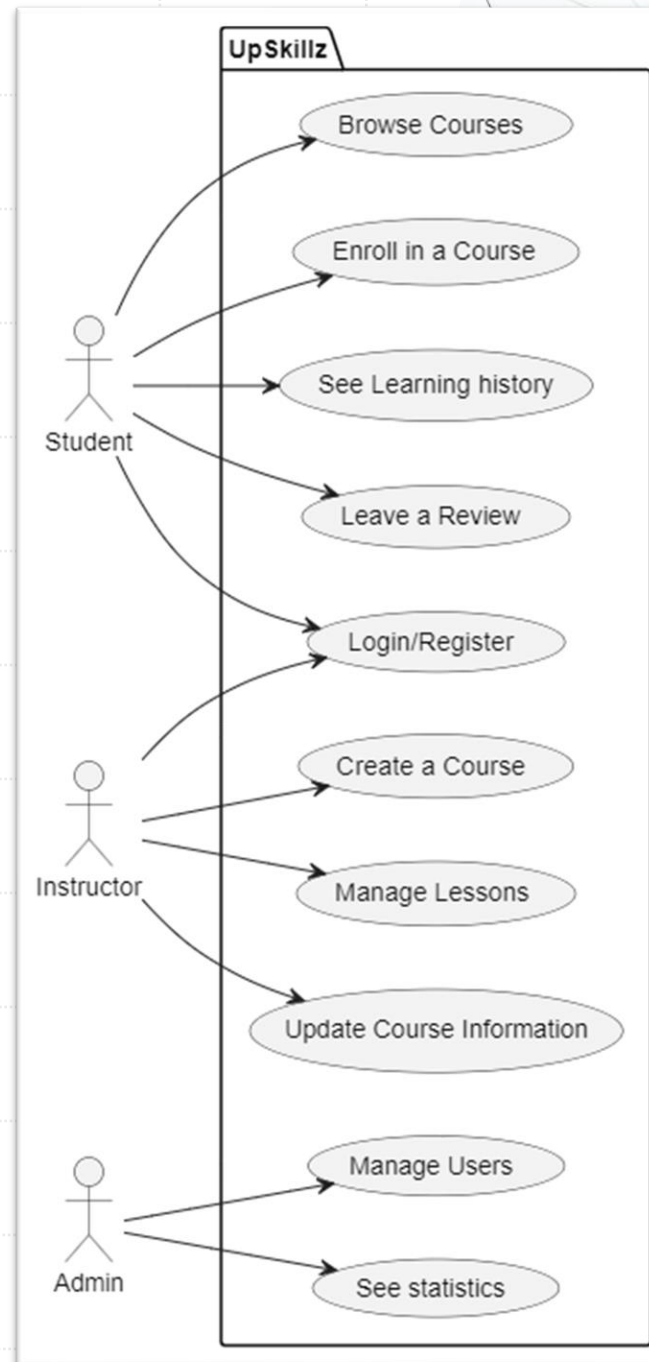
Enables instructors to:

- Manage courses and lesson content efficiently.
- Monitor student engagement and performance.

Project Solution

Use Case Diagram

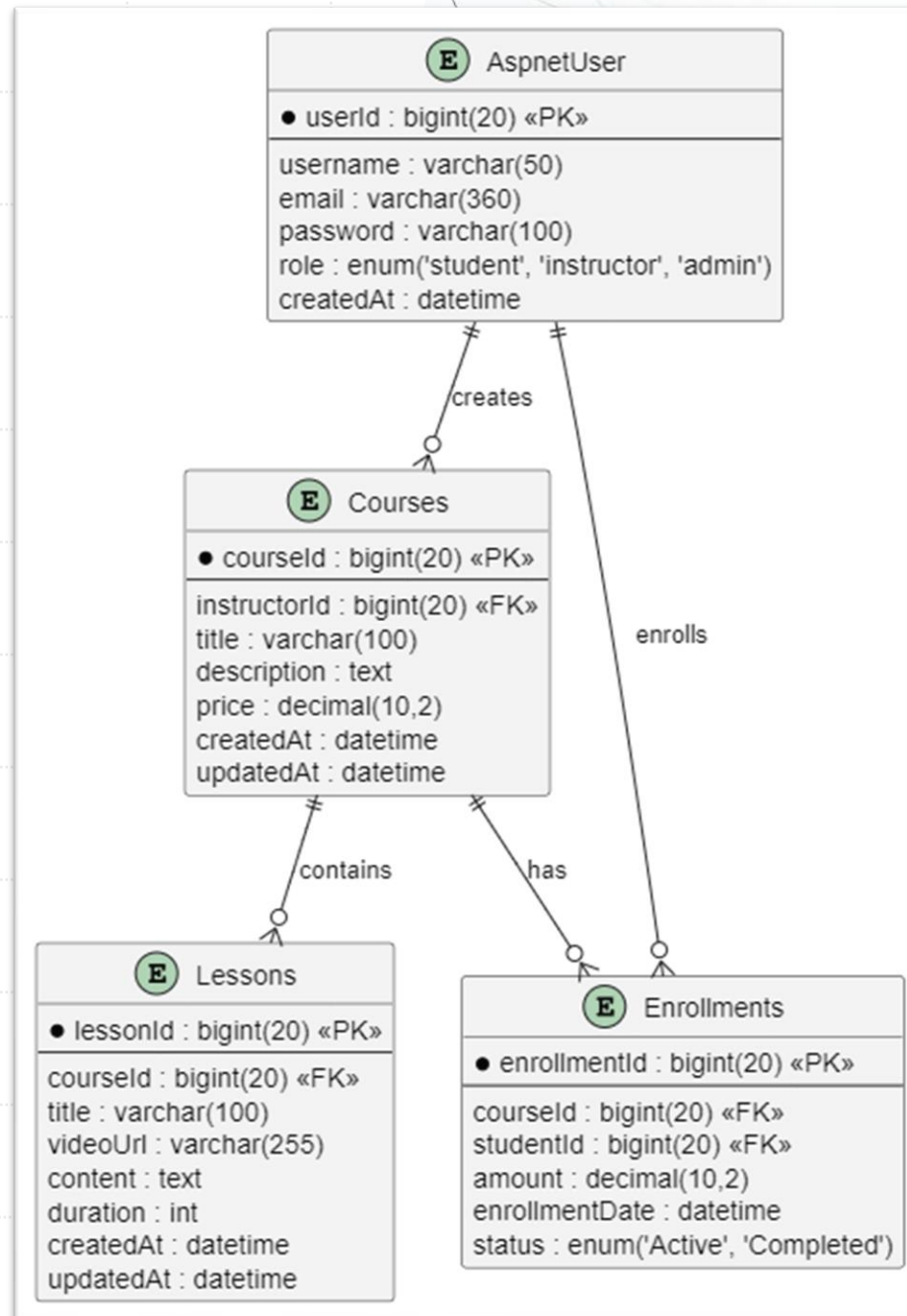
- Student
- Instructor
- Admin

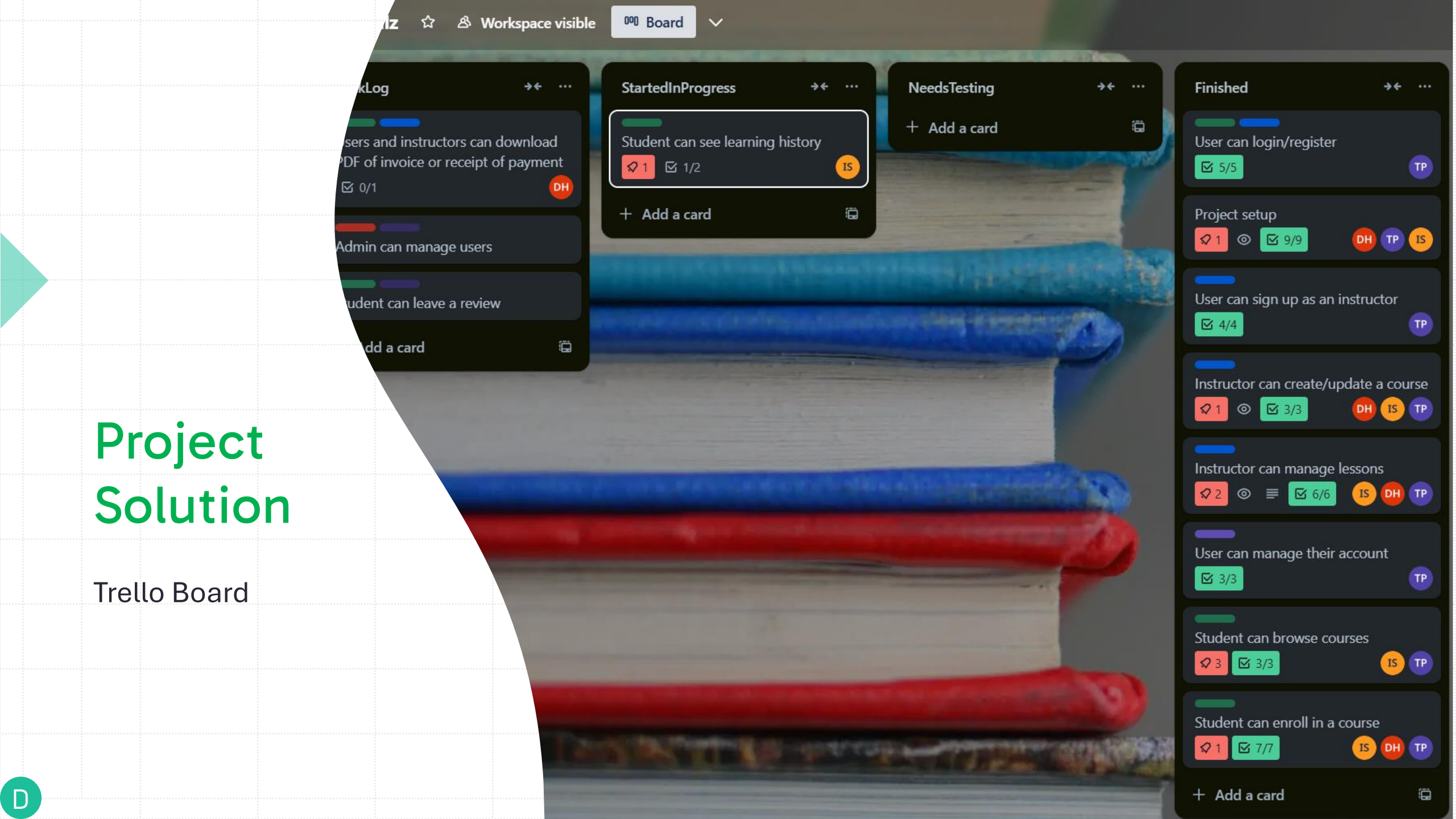


Project Solution

ER Diagram

- Users
- Courses
- Lessons
- Enrollements





Project Solution

Trello Board

Project Solution

GitHub

- Individual branches
- Product for deployment branches

← Build and deploy ASP.Net Core app to Azure Web App - upskillz

✓ Merge pull request #26 from hdcola/tamara #3

Summary

Jobs

- ✓ build
- ✓ deploy

Run details

- Usage
- Workflow file

Triggered via push 1 hour ago

hdcola pushed 1734c90 product Success

Total duration 2m 48s

CD.yml

on: push

build 51s → deploy 1m 40s

https://upskillz-gzcvddg2gwcwchcd.ca...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	9 minutes ago				Default

Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. [Learn more](#)

Dismiss Protect this branch

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
product	1 hour ago	✓ 2 / 2	3	0	
danny	2 hours ago		20	0	#21

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
iana	8 minutes ago		0	2	#23
tamara	38 minutes ago		1	0	#24
product	1 hour ago	✓ 2 / 2	3	0	
danny	2 hours ago		20	0	#21

Checkout form

Payment

Credit card information

Card number

MM / YY CVC

Pay

Your cart

How to drive your Tesla

Trust me, when you buy a Tesla, you are not going to drive it. Since you spent tens of thousands of dollars on a Tesla, then you must buy this course and learn it so you can actually drive your Tesla, otherwise you won't even be able to open its doors!

Instructor: danny

Total (USD)

stripe-dotnetPublic

Watch81Fork671Star1.4k

master14 Branches798 Tags

Go to fileAdd fileCode

jar-stripeBump version to 47.2.0e78194c · 19 minutes ago3,369 Commits

.config	Add coveralls step	2 years ago
.github	Stop using deprecated versions of actions (#2965)	3 months ago
.vscode	update recommended extensions and settings (#2924)	6 months ago
src	Bump version to 47.2.0	19 minutes ago
.editorconfig	added ConfigureAwait(false) calls to await statements, to ...	2 months ago
.gitattributes	Update .gitattributes (#2669)	last year
.gitignore	Add support for to Issuing Transaction	5 years ago
CHANGELOG.md	Bump version to 47.2.0	19 minutes ago
CODE_OF_CONDUCT.md	Contributor Covenant (#1820)	5 years ago
LICENSE	Revert "Fix license packaging"	6 years ago
Makefile	Fix user agent in .net sdks (#3012)	2 months ago
OPENAPI_VERSION	Update generated code (#3036)	33 minutes ago

About

Stripe.net is a sync/async .NET 4.6.1+ client, and a portable class library for stripe.com.

Readme

Apache-2.0 license

Code of conduct

Security policy

Activity

Custom properties

1.4k stars

81 watching

571 forks

Report repository

Releases780

v47.2.0Latest19 minutes ago

+ 779 releases

Project Solution

Payment with Stripe

Find anything

SETTING UP

Introduction

Including Stripe.js

Initializing Stripe.js

THE ELEMENTS OBJECT

Create an Elements object

Create an Elements object without an Intent

Update an Elements object

Update-end event

Fetch Elements Updates

Stripe.js reference

This reference documents every object and method available in Stripe's browser-side JavaScript library, Stripe.js. Use our [React Stripe.js reference](#) if you want to add Elements to your React based app.

You can use Stripe.js' APIs to tokenize customer information, collect sensitive payment details using customizable [Stripe Elements](#), and accept payments with [browser payment APIs](#) like Apple Pay and the Payment Request API.

Project Solution

Payment with Stripe

```
19 0 references
20 public CheckoutController(ApplicationDbContext context, ILogger<CheckoutController> logger, StripeService stripeService)
21 {
22     _context = context;
23     _logger = logger;
24     _stripeService = stripeService;
25     _stripePublicKey = Environment.GetEnvironmentVariable("Stripe_PublicKey") ?? string.Empty;
26 }
27
28 // GET: Checkout
29 0 references
30 public IActionResult Index(string cid)
31 {
32     if (!int.TryParse(cid, out int courseId))
33     {
34         _logger.LogError("Invalid course ID: {CourseId}", cid);
35         return BadRequest("Invalid course ID");
36     }
37
38     _logger.LogInformation("Accessing checkout page for course ID {CourseId}", courseId);
39     ViewBag.PublishableKey = _stripePublicKey;
40     ViewBag.CourseId = courseId;
41
42     var course = _context.Courses
43         .Include(c => c.Instructor)
44         .FirstOrDefault(c => c.CourseId == courseId);
45     if (course == null)
46     {
47         _logger.LogError("Course not found for ID: {CourseId}", courseId);
48         return NotFound("Course not found");
49     }
50
51     var userName = User.Identity?.Name ?? "Guest";
52     var email = User.FindFirstValue(ClaimTypes.Email);
53     ViewBag.UserName = userName;
54     ViewBag.Email = email;
55
56     if (course.Price == 0)
57     {
58         return View("Free", course);
59     }
60
61     return View(course);
62 }
```

```
60 <form id="payment-form">
61     <h4 class="mb-3">Payment</h4>
62     <label for="cc-number" class="form-label">Credit card information</label>
63     <div id="card-element">
64         <!-- A Stripe Element will be inserted here. -->
65     </div>
66     <hr class="my-4">
67
68     <div id="payment-result" class="alert d-none mb-3"></div>
69
70     <button class="w-100 btn btn-primary btn-lg" type="submit">Pay</button>
71     <div id="continue"></div>
72 </form>
73 </div>
74 </div>
75 </main>
76
77 <script src="https://js.stripe.com/v3/"></script>
78 <script>
79     const stripe = Stripe('@ViewBag.PublishableKey');
80     const elements = stripe.elements();
81     const cardElement = elements.create('card');
82     cardElement.mount('#card-element');
83
84     const userName = '@ViewBag.UserName';
85     const email = '@ViewBag.Email';
86     const amount = @((int)(Model.Price * 100)); // Convert to cents
87     const courseId = @Model.CourseId;
88
89     const form = document.getElementById('payment-form');
90     const resultContainer = document.getElementById('payment-result');
91
92     form.addEventListener('submit', async (event) => {
93         event.preventDefault();
94
95         // Show loading state
96         const submitButton = form.querySelector('button');
97         submitButton.disabled = true;
98         submitButton.innerHTML = '<span class="spinner-border spinner-border-sm me-2"></span>Processing...';
99
100         const response = await fetch('/Checkout/CreatePaymentIntent', {
101             method: 'POST',
102             headers: { 'Content-Type': 'application/json' },
103             body: JSON.stringify({
104                 Name: userName,
105                 Email: email,
106                 Amount: amount,
107                 CourseId: courseId
108             })
109         });
110         const { clientSecret } = await response.json();
111     });
112 }
```


Project Solution

Payment with Stripe

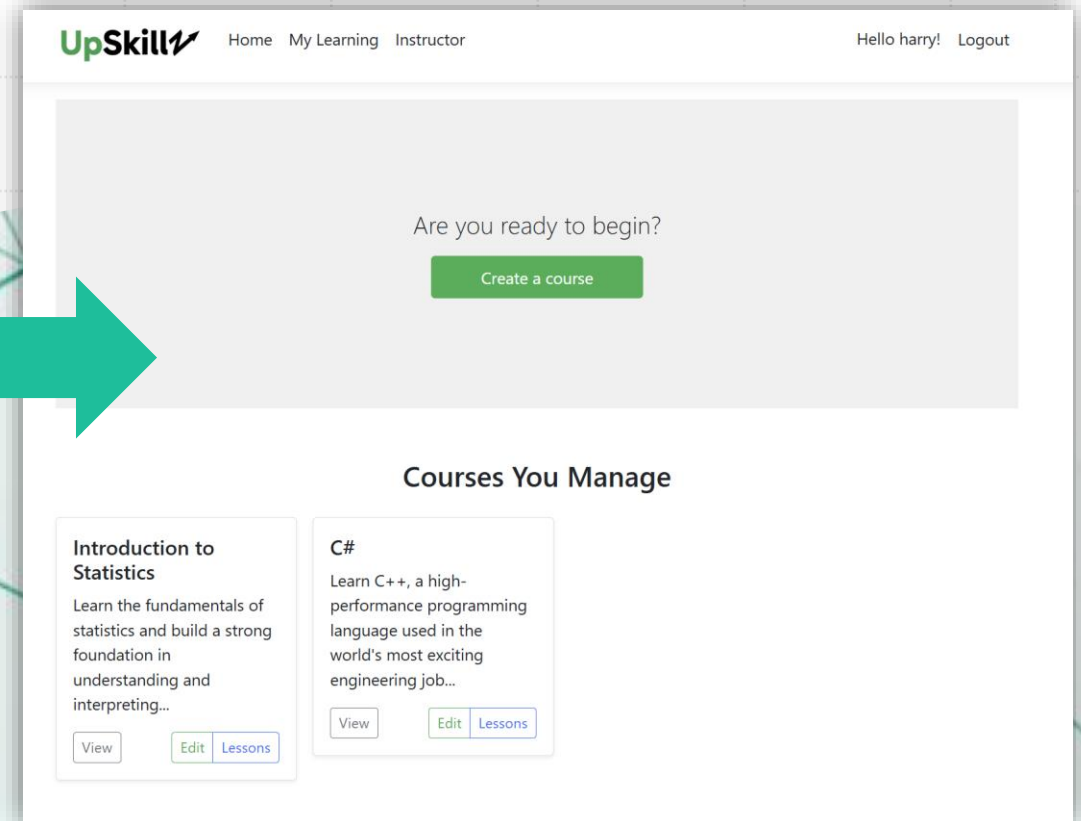
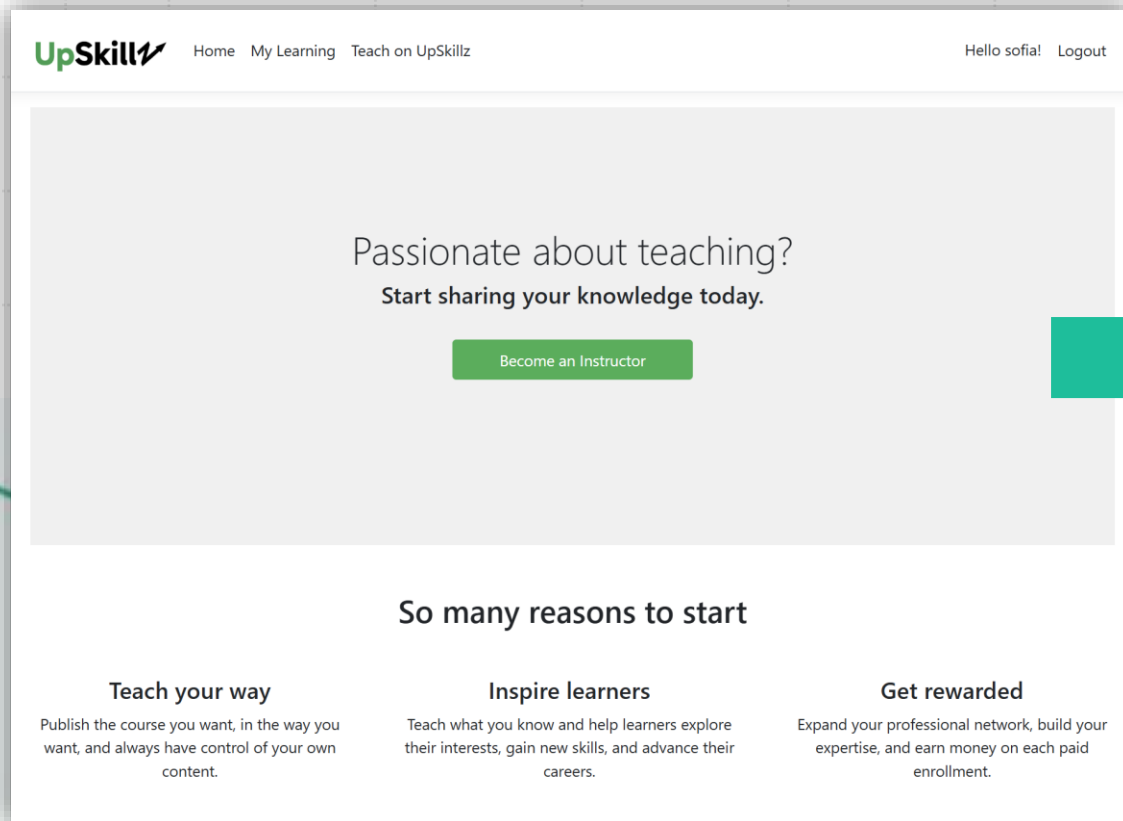
```
62 [HttpPost]
63 0 references
64 public IActionResult CreatePaymentIntent([FromBody] PaymentModel paymentModel)
65 {
66     try
67     {
68         _logger.LogInformation("Creating payment intent for amount: {Amount}", paymentModel.Amount);
69         var paymentIntent = _stripeService.CreatePaymentIntent(paymentModel.Amount);
70         _logger.LogInformation("Payment intent created successfully. ID: {PaymentIntentId}", paymentIntent.Id);
71
72         // Get the current user
73         var userId = User.FindFirstValue(ClaimTypes.NameIdentifier);
74         var student = _context.Users.FirstOrDefault(u => u.Id == userId);
75
76         // Get the course
77         var course = _context.Courses.FirstOrDefault(c => c.CourseId == paymentModel.CourseId);
78
79         if (course == null)
80         {
81             _logger.LogError("Course not found for ID: {CourseId}", paymentModel.CourseId);
82             return NotFound("Course not found");
83         }
84
85         if (student == null)
86         {
87             _logger.LogError("User not found for ID: {UserId}", userId);
88             return NotFound("User not found");
89         }
90
91         // Create a new enrollment
92         var enrollment = new Enrollment
93         {
94             Student = student,
95             Course = course,
96             Amount = course.Price, // Convert cents to decimal
97             EnrollmentDate = DateTime.Now,
98             Status = EnrollmentStatus.Active,
99             PaymentReference = paymentIntent.Id
100         };
101
102         // Save the enrollment to the database
103         _context.Enrollments.Add(enrollment);
104         _context.SaveChanges();
105
106         return Json(new { clientSecret = paymentIntent.ClientSecret });
107     }
108 }
```

```
4 4 references
5 public class StripeService
6 {
7     2 references
8     private readonly string _secretKey;
9
10    0 references
11    public StripeService(IConfiguration configuration)
12    {
13        _secretKey = Environment.GetEnvironmentVariable("Stripe__SecretKey") ?? string.Empty;
14        StripeConfiguration.ApiKey = _secretKey;
15    }
16
17    1 reference
18    public PaymentIntent CreatePaymentIntent(long amount, string currency = "usd")
19    {
20        var options = new PaymentIntentCreateOptions
21        {
22            Amount = amount,
23            Currency = currency,
24            PaymentMethodTypes = new List<string> { "card" }
25        };
26
27        var service = new PaymentIntentService();
28        return service.Create(options);
29    }
30 }
```

Project Solution

Instructor

Users can become Instructors -> Create and manage courses



Project Solution

Lessons

- Keep track of completed lessons
- Lessons have embedded videos

Home / Courses / Introduction to Statistics / Lessons

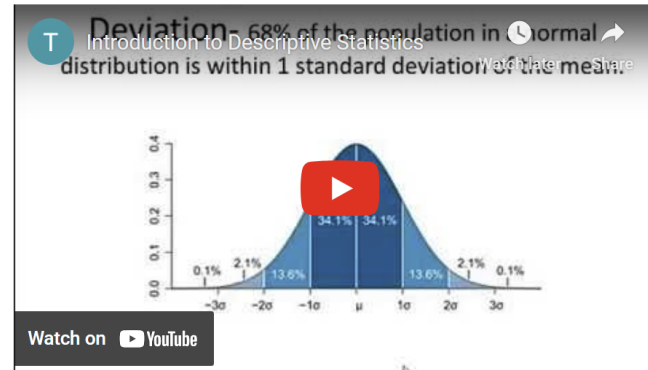
Introduction to Statistics

Understanding Descriptive Statistics ✓

Probability Theory Basics

Introduction to Inferential Statistics

Understanding Descriptive Statistics



Descriptive statistics is the foundation of data analysis. In this lesson, you'll learn how to summarize and describe the key features of a dataset. You'll explore measures such as the mean, median, mode, range, and standard deviation. These concepts help statisticians and data analysts make sense of data by transforming complex datasets into meaningful summaries. You will also dive into visual representations of data, like histograms, bar charts, and box plots, which make it easier to observe trends and patterns. Understanding descriptive statistics allows you to effectively communicate insights and trends about data to a broader audience.

Completed

Project solution

Your account

- Managing profile details
- History of payments

Manage your account

Profile

Email

Password

Purchase History

Profile

Username

sofia





Save

Manage your account

Profile	Purchase History			
Email	Course	Date	Amount	Payment Reference
Password	Introduction to Statistics	17 December 2024	\$0.00	Free
Purchase History	Blockchain Developer	18 December 2024	\$0.00	Free
	How to Cook Chinese food	18 December 2024	\$999.99	pi_3QXXLGRKj8j7VclL12qB8mpl
	.NET	18 December 2024	\$4.99	pi_3QXYKHRKj8j7VclL1hBr3kTW

Project solution

Search courses


Web Forms	MVC	Web API	ASP.NET Core MVC
ASP.NET 4.6 (System.Web)		ASP.NET Core 1.0 (Microsoft.AspNetCore)	
.NET Framework Platform for .NET applications on Windows 		.NET Core Cross-platform, modular libraries & runtime optimized for server and cloud workloads   	

Jelvix

AspNet Core

A Full course for beginner

[View](#)



.NET

A Full course for beginners

[View](#)

Challenges and Solutions

ASP.NET MVC Pattern

```
▼ Models
C# Course.cs
C# Enrollment.cs
C# EnrollmentModel.cs
C# Lesson.cs
C# PaymentModel.cs
C# StudentLesson.cs
C# UploadModel.cs
C# User.cs
```

```
▼ Views
  > Blob
  > Checkout
  > Courses
  > Dashboard
  > Home
  > Lessons
  > Shared
```

```
▼ Views
  ▼ Blob
    _ViewImports.cshtml
    Index.cshtml
  ▼ Checkout
    _ViewImports.cshtml
    Free.cshtml
    Index.cshtml
  ▼ Courses
    _ViewImports.cshtml
    Cards.cshtml
    Course.cshtml
    Delete.cshtml
    Details.cshtml
    Edit.cshtml
    Explore.cshtml
    Index.cshtml
  ▼ Dashboard
    _ViewImports.cshtml
    Course.cshtml
    Enrolled.cshtml
    Index.cshtml
  ▼ Home
    _ViewImports.cshtml
    Index.cshtml
  ▼ Lessons
    _ViewImports.cshtml
```

```
▼ Controllers
C# BlobController.cs
C# CheckoutController.cs
C# CoursesController.cs
C# DashboardController.cs
C# HomeController.cs
C# LessonsController.cs
```


```
▼ Services
C# BlobStorageService.cs
C# StripeService.cs
```

Challenges and Solutions

ASP.NET MVC Pattern


Partial views

@await Html.PartialAsync("~/Views/Courses/Cards...")



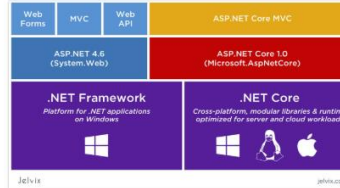
Introduction to Statistics
Learn the fundamentals of statistics and build a strong foundation in understanding and interpreting...

[View](#)



Blockchain Developer
Master blockchain, from fundamentals to smart contracts. In the first part of this program, you'll l...


[View](#)



ASP.NET Core MVC
ASP.NET 4.6 (System.Web) | ASP.NET Core 1.0 (Microsoft.AspNetCore)
.NET Framework | .NET Core
Platform for .NET applications on Windows | Cross-platform, modular libraries & runtime optimized for server and cloud workloads


Asp.Net Core
A Full course for beginner

[View](#)




.NET
A Full course for beginners

[View](#)



C#
Learn C++, a high-performance programming language used in the world's most exciting engineering job...

[View](#)



How to Cook Chinese food
Make it for your loved people, grab their stomachs, own their hearts

[View](#)

```
<section class="py-5 text-center container">
  <div class="row py-lg-2">
    <div class="col-lg-6 col-md-8 mx-auto">
      <h1 class="fw-light text-upskillz">Unlock Your Potential with UpSkillz</h1>
      <p class="lead text-body-secondary">Gain expertise in the latest technologies,
        learn industry-recognized certifications, and take your career to new heights.</p>
      <p>
        <a asp-controller="Courses" asp-action="Explore" class="btn btn-upskillz my-2 text-white">Explore Courses</a>
      </p>
    </div>
  </div>
</section>

@await Html.PartialAsync("~/Views/Courses/Cards.cshtml", Model)
```

EXPLORER

- UPSKILLZ
 - Views
 - Cards.cshtml

```
Views > Courses > Cards.cshtml
1 @model IEnumerable<Upskillz.Models.Course>
2
3 <div class="album py-2 bg-body-tertiary">
4   <div class="container">
5     <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">
6       @if (Model != null && Model.Any())
7       {
8         @foreach (var course in Model)
9         {
10          <div class="col">
11            <div class="card shadow-sm d-flex flex-column h-100">
12              
15              <div class="card-body d-flex flex-column flex-grow-1">
16                <h5 class="card-title">@course.Title</h5>
17
18                @{
19                  var maxLength = 100; // Number of characters to show
20                  var description = course.Description;
21
22                  if (description.Length > maxLength)
23                  {
24                    description = description.Substring(0, maxLength) + "...";
25                  }
26                }
27                <p class="card-text">@description</p>
28              <div class="mt-auto">
```

What we learned

ASP.NET MVC Pattern

Flexible and Efficient Routing:

asp-controller="**Courses**"

asp-action="**Details**"

asp-route-id="**@courseId**"



Introduction to Statistics

Learn the fundamentals of statistics and build a strong foundation in understanding and interpreting...

[View](#)

[Home](#) / [Courses](#) / [Introduction to Statistics](#)

Price: \$ 45.00

Introduction to Statistics

Instructor: harry



[View the course](#)

3 learners enrolled

Course Description

Learn the fundamentals of statistics and build a strong foundation in understanding and interpreting data! This course covers essential concepts such as measures of center, including mean, median, and mode, as well as measures of spread like range, variance, and standard deviation. You'll gain a clear understanding of how these statistical tools summarize data and provide insights into real-world applications, from business decision-making to scientific research. Best of all, this course is designed for anyone - no coding experience is required. Dive into the fascinating world of probability and discover how to analyze and interpret random events. You'll explore probability distributions, including the normal, binomial, and Poisson distributions, which are fundamental to predicting outcomes and understanding trends. By mastering these concepts, you'll be equipped to tackle problems involving uncertainty with confidence and clarity. The course emphasizes conceptual understanding and practical applications, making it accessible and relevant for learners from all backgrounds. Conclude your journey by learning the basics of hypothesis testing, a powerful statistical method used to make informed decisions based on data. Understand concepts such as null and alternative hypotheses, p-values, and confidence intervals, which are key to validating claims and drawing conclusions. Whether you're preparing for further studies, improving your data literacy, or enhancing your professional skills, this course will empower you with the tools needed to approach data critically and effectively.

Understanding Descriptive Statistics

Descriptive statistics is the foundation of data analysis. In this lesson, you'll learn how to summa...

Probability Theory Basics

In this lesson, we introduce basic probability concepts, which are essential in making informed pred...

Introduction to Inferential Statistics

In this lesson, you'll explore inferential statistics, which allows us to make conclusions about a p...


```
// Register User as instructor
await _userManager.AddToRoleAsync(user, "Instructor");
// Update cookie so it is aware of new role
await _signInManager.RefreshSignInAsync(user);
```

```
// GET: Courses/Create
[Authorize(Roles = "Instructor")]
0 references
public async Task<IActionResult> Course(int? id)
{
    var course = new Course();
    if (id != null)
    {
```

What we learned

User Roles

- You need to refresh the signin cookie to reflect the changes in Roles.

What we learned

User Roles

- Frontend Role validation and authentication.

```
<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
  <ul class="navbar-nav flex-grow-1">
    <li class="nav-item">
      <a class="nav-link text-dark" asp-controller="Home" asp-action="Index">Home</a>
    </li>
    @if (SignInManager.IsSignedIn(User))
    {
      <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Dashboard" asp-action="Enrolled">My
          Learning</a>
      </li>
    }
    @if (User.IsInRole("Instructor"))
    {
      <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Dashboard" asp-action="Index">Instructor</a>
      </li>
    }
    else
    {
      <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Dashboard" asp-action="Index">Teach on
          UpSkillz</a>
      </li>
    }
  </ul>
  <partial name="_LoginPartial" />
  @*
```

What we learned

Loading Partial Views in Ajax to enable search results

```
$(`form#search-courses`).on("submit", function (event) {
    event.preventDefault();
    const searchTerms = $("#search").val();

    $.ajax({
        type: "POST",
        url: '/Courses/Search',
        contentType: "application/json; charset=utf-8",
        headers: { "RequestVerificationToken": $('input[name="__RequestVerificationToken"]').val() },
        data: JSON.stringify({ SearchTerms: searchTerms }),
        success: function (data) {
            $("#results").html(data);
        },
        error: function (req, err) {
            console.log("error", err);
        }
    });
});
```

```
// POST: Courses/Explore/Search
[HttpPost]
[AllowAnonymous]
0 references
public async Task<IActionResult> Search([FromBody] Search search)
{
    _logger.LogInformation($"Search courses {search.SearchTerms}");

    if (string.IsNullOrEmpty(search.SearchTerms))
    {
        _logger.LogInformation("Search cannot be null.");
        return BadRequest("Search cannot be null.");
    }

    var courses = await _context.Courses
        .Where(c => c.Title
        .Contains(search.SearchTerms))
        .ToListAsync();

    return PartialView("~/Views/Courses/Cards.cshtml", courses);
}
```

```
[HttpPost]
[AllowAnonymous]
0 references
public async Task<IActionResult> SearchReset()
{
    var courses = await _context.Courses.ToListAsync();
    return PartialView("~/Views/Courses/Cards.cshtml", courses);
}
```

What we learned

__RequestVerificationToken

```
$('.complete-lesson-btn').on('click', function (e) {  
    e.preventDefault();  
  
    var lessonId = $(this).data('lesson-id');  
    var token = $('input[name="__RequestVerificationToken"]').val();  
  
    $.ajax({  
        url: '/Lessons/CompleteLesson/' + lessonId,  
        type: 'POST',  
        headers: {  
            "RequestVerificationToken": token  
        },  
        success: function (response) {  
            location.reload();  
        },  
        error: function (xhr) {  
            console.error(xhr.responseText);  
            alert('An error occurred while completing the lesson.');        }  
    });  
});
```

```
Lessons > LessonsController.cs > LessonsController > CompleteLesson  
public class LessonsController : Controller  
{  
    public async Task<IActionResult> CompleteLesson(int id)  
    {  
        var studentLesson = await _context.StudentsLessons  
            .FirstOrDefaultAsync(sl => sl.LessonId == id && sl.UserId == userId);  
  
        _logger.LogInformation($"studentLesson: {studentLesson}");  
  
        if (studentLesson == null)  
        {  
            studentLesson = new StudentLesson
```

Home / Courses / Introduction to Statistics / Lessons

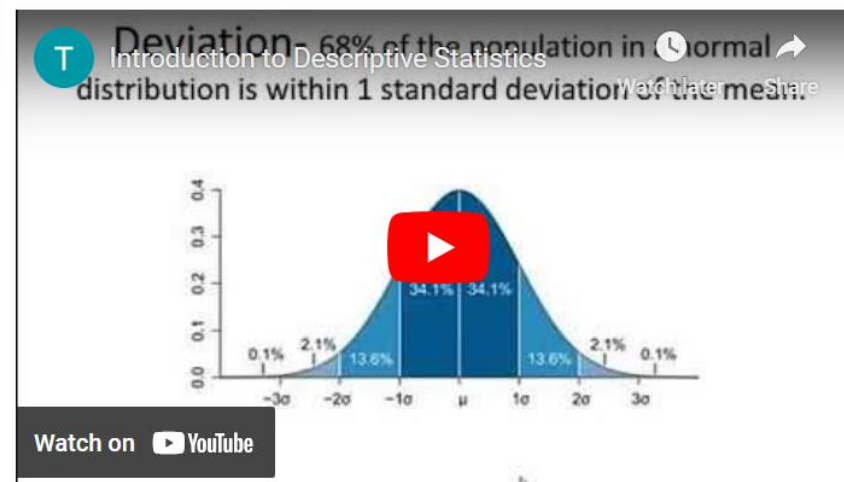
Introduction to Statistics

Understanding Descriptive Statistics

Probability Theory Basics

Introduction to Inferential Statistics

Understanding Descriptive Statistics



Descriptive statistics is the foundation of data analysis. In this lesson, you'll learn how to summarize and describe the key features of a dataset. You'll explore measures such as the mean, median, mode, range, and standard deviation. These concepts help statisticians and data analysts make sense of data by transforming complex datasets into meaningful summaries. You will also dive into visual representations of data, like histograms, bar charts, and box plots, which make it easier to observe trends and patterns. Understanding descriptive statistics allows you to effectively communicate insights and trends about data to a broader audience.

Completed

Dear

Thank you for your purchase.

RECEIPT

Coursera Inc.
381 E. Evelyn Avenue
Mountain View, CA 94041 USA

04-19-2021
Order Number: 122481033

Product Details	Price	Qty
Course Certificate for Estatística não-paramétrica para a tomada de decisão	USD 29.00	1

TOTAL (USD): 29.00

Please keep this receipt as record of your payment.

View our [refund policy](#).

View your [purchase history](#).

Happy Learning!

Future Work

- Google Authentication
- PDF invoices
- Administrator role
- User reviews



Summary

Fully functional application for:

- Explore and search the courses
- View a course details
- Enroll and pay for the course
- Showing purchased courses through My Learning
- Mark your progress in the course
- Create a course for Instructor
- Manage lessons in Course
- Continuous Deployment to Azure Web App