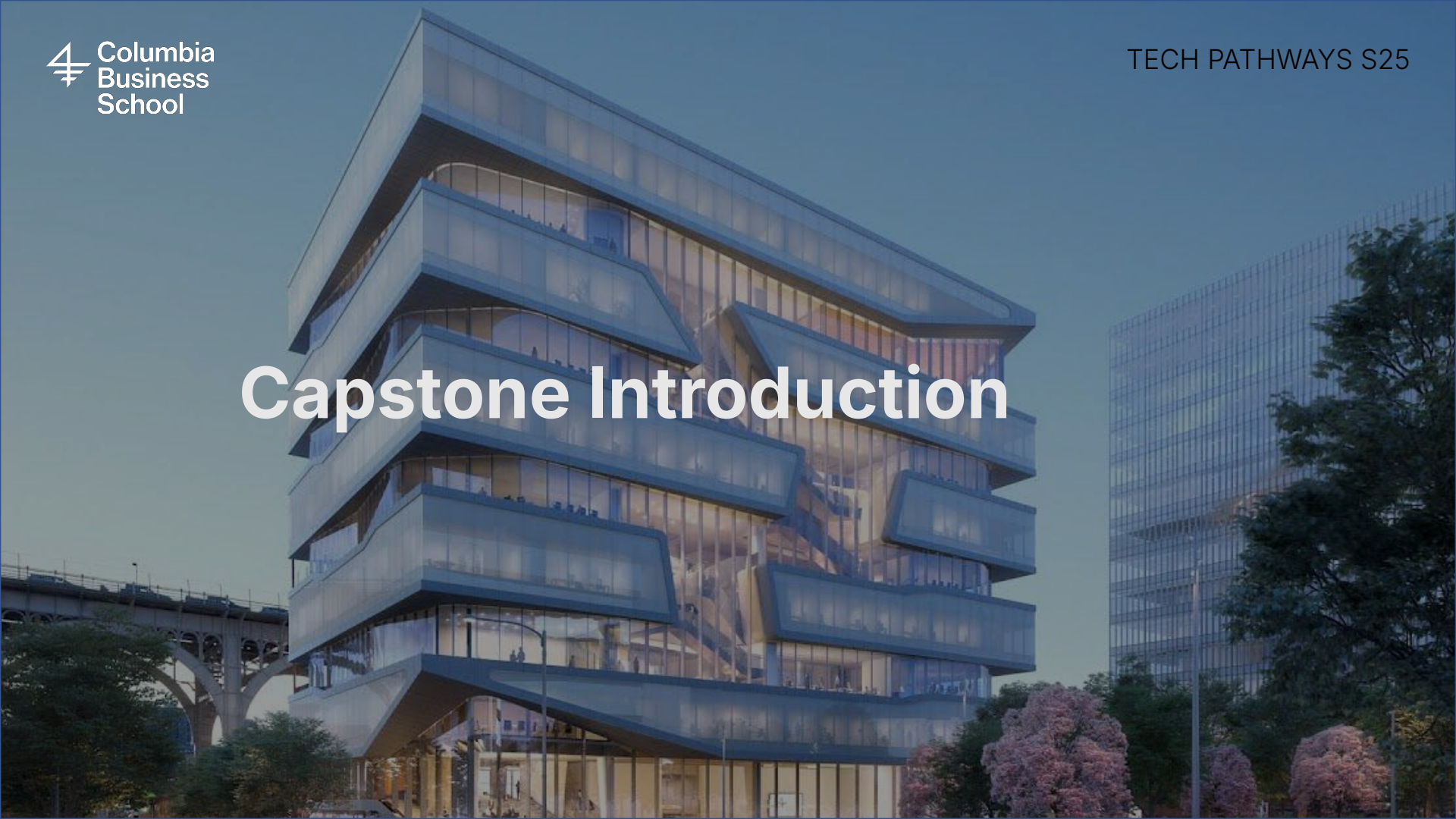
Columbia Business School

TECH PATHWAYS S25

# Capstone Introduction



# Today's Lesson - Capstone Introduction

**Focus Concepts**

- Understanding the structure and requirements of the Weather Dashboard Capstone Project
- Exploring the "choose-your-own-adventure" approach to project features
- Recognizing the balance between individual work and optional team collaboration
- Learning how to evaluate and select appropriate features based on skill level and interests
- Understanding how the capstone demonstrates mastery of course concepts

**Learning Objectives**

- Explain the overall structure of the Weather Dashboard Capstone Project
- Identify the required core components and optional features
- Select three features from the menu that align with their skills and interests
- Understand the timeline and milestones for the 7-week project
- Set up their development environment for the capstone project
- Create a personal vision for their weather application

# Today’s Lesson - Capstone Introduction

**Key Terms**

- Capstone Project: A culminating project that demonstrates mastery of course concepts
- Core Component: The required foundation that every fellow must implement
- Feature Menu: A selection of optional features categorized by difficulty (⭐ to ⭐⭐⭐)
- Individual Component: The 70% of the project completed independently
- Team Component: The optional 30% collaborative feature
- Project Owner: The team member who manages the GitHub repository and integration
- API Integration: Connecting to external services to fetch weather data
- Enhancement: Personal touches to make the application unique
- Minimum Viable Product (MVP): The basic working version with core functionality
- Feature Integration: The process of adding new features to the core application

# Part 1: Project Overview and Structure

# Project Structure Overview - Require Core

**Required Core (Everyone will build these pieces)**

- Basic weather data fetching via API
- Simple Tkinter GUI displaying current weather
- File-based data storage
- Error handling

"This core ensures everyone has a working weather application by the end of Week 12. It's your foundation - like the frame of a house."

# Project Structure Overview - Choose Your Features (Pick 3)

**Choose Your Features (Pick 3)**

- 12 features available across 4 categories
- Difficulty ratings from ⭐ (beginner) to ⭐⭐⭐ (advanced)
- Each feature is self-contained
- Build one feature per week (Weeks 13-15)

"This is where your app becomes unique. Choose features that interest you or challenge you appropriately."

# Project Structure Overview - Personal Enhancements (Pick 1)

**Personal Enhancement (Pick 1)**

- Add personality to your application
- Creative freedom within technical constraints
- Makes your app memorable

"This could be a weather mascot, custom descriptions, sound effects, or anything that adds character."

# Project Structure Overview - Optional Team Component

**Optional Team Component**

- Simple collaborative feature (Week 15)
- Requires 3-4 team members
- Uses GitHub for coordination
- Worth 30% of grade (or do extra individual work)

"Teams aren't formed until Week 13, so don't worry about this yet."

# Project Structure Overview - Data Features

**Data Features ⭐**

1. Weather History Tracker
    - Save daily weather to CSV
    - Display last 7 days
    - Calculate weekly averages
2. Simple Statistics
    - Min/max temperature tracking
    - Weather type counting
    - Display in labels
3. City Comparison
    - Compare 2 cities side-by-side
    - Show temperature differences
    - Simple text display

# Project Structure Overview - Visual Features

**Visual Features ⭐⭐**

1. Temperature Graph
    - Line graph of temperature history
    - Matplotlib embedded in Tkinter
    - Use provided template
2. Weather Icons
    - Canvas-based weather representations
    - Color-coded conditions
    - Simple animations
3. Theme Switcher
    - Day/night modes
    - Weather-based colors
    - User preferences

# Project Structure Overview - Interactive Features

**Interactive Features ⭐⭐**

1. Weather Journal
    - Daily weather notes
    - Mood tracking
    - Text file storage
2. Favorite Cities
    - Save preferred locations
    - Quick switching
    - Persistent storage
3. Weather Alerts
    - Temperature thresholds
    - Simple notifications
    - User settings

# Project Structure Overview - Smart Features

**Smart Features ⭐⭐⭐**

1. Tomorrow's Guess
    - Basic prediction logic
    - Confidence levels
    - Accuracy tracking
2. Trend Detection
    - Temperature trend arrows
    - Pattern identification
    - Simple analysis
3. Activity Suggester
    - Weather-based recommendations
    - Custom activity lists
    - Random suggestions

# Project Structure Overview - Timeline

Timeline Overview:

- Week 11 (This week): Planning and setup
- Week 12: Build core functionality
- Week 13: Implement first feature + form teams
- Week 14: Implement second feature
- Week 15: Implement third feature + team component
- Week 16: Enhancement, documentation, polish
- Week 17: Presentations and demos

# Breakout #1 - Feature Selection & Project Visioning

Activity: In small groups, explore the feature menu and begin planning your personal weather dashboard.

Instructions:

1. Breakout groups will be your normal groups
2. Each person should:
    - Review all 12 features in detail
    - Identify 4-5 features that interest them
    - Consider their current skill level
    - Think about time commitment
3. Discuss your choices with the group:
    - Why did you choose these features?
    - What challenges do you anticipate?
    - How will these features work together?
4. Help each other make final selections

# Breakout #1 - Feature Selection & Project Visioning Review

Activity: In small groups, explore the feature menu and begin planning your personal weather dashboard.

Expected Output:

- Each fellow has a preliminary list of 3 features
- Notes on why these features were chosen
- Identification of potential challenges
- Initial ideas for personal enhancement

Discussion Points:

- How do your chosen features complement each other?
- What skills will you need to develop?
- How will these features make your app unique?
- What's your backup plan if a feature proves too difficult?

# Break #1

# Part 2: Core Component Deep Dive - Live Coding

# Breakout #2 - Setting Up Development Environment

Activity: In small groups, set up your development environment and test the API connection.

Instructions:

1. Create a new project folder: `WeatherDashboard`
2. Set up the following structure:
   WeatherDashboard/ ├── main.py ├── config.py ├── data/ ├── docs/ └── tests/
3. Sign up for OpenWeatherMap API key (or use provided key)
4. Create a basic Tkinter window to ensure GUI library is working
5. Test the API with a simple request:
   import requestsapi_key = "your_key_here"city = "Seattle"url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"response = requests.get(url)print(response.json())

# Breakout #2 - Setting Up Development Environment

Activity: In small groups, set up your development environment and test the API connection.

Expected Output:

- Working project folder structure
- Successful API call returning weather data
- Basic Tkinter window displaying
- Notes on any setup challenges

Discussion Points:

- Did everyone successfully connect to the API?
- What data fields look most useful from the API response?
- How might you handle API errors gracefully?
- What additional data might you want to store?

Columbia Business School
AT THE VERY CENTER OF BUSINESS

# Break #2

# Part 3: Project Planning and Success Strategies

# Project Planning - Roadmap - Weekly Milestones

**Week 11 (This Week):**

- Monday: Project introduction and feature selection ✓
- Tuesday: System architecture planning
- Wednesday: Data modeling design
- Thursday: Create detailed project timeline
- Friday: Set up testing framework

**Week 12:**

- Monday-Tuesday: Implement core API functionality
- Wednesday: Build basic GUI
- Thursday: Add file storage
- Friday: Testing and debugging

# Project Planning - Roadmap - Weekly Milestones

**Week 13:**

- Monday-Tuesday: Implement Feature #1
- Wednesday: Integration with core
- Thursday: Testing
- Friday: Team formation and planning

**Week 14:**

- Feature #2 implementation
- Begin team repository setup

# Project Planning - Roadmap - Weekly Milestones

**Week 15:**

- Feature #3 implementation
- Team feature development

**Week 16:**

- Personal enhancement
- Documentation
- Polish and refinement

**Week 17:**

- Final testing
- Presentation preparation
- Demos

# Project Planning - Success Strategies

- **Start Simple, Build Up**
  - Get the core working first
  - Add features incrementally
  - Test after each addition
- **Use Version Control**
  - Commit after each working feature
  - Use descriptive commit messages
  - Create branches for experiments
- **Document As You Go**
  - Comment your code immediately
  - Keep a development journal
  - Screenshot interesting milestones

# Project Planning - Success Strategies

- **Ask for Help Early**
  - Don't struggle alone for hours
  - Share specific error messages
  - Use office hours effectively
- **Plan for Problems**
  - Budget extra time for debugging
  - Have backup plans for features
  - Keep offline data for testing

# Project Planning - Common Pitfalls to Avoid

- **Feature Creep**
  - Stick to your chosen 3 features
  - Complete before adding extras
  - Resist the urge to over-engineer
- **Perfectionism**
  - "Done" is better than "perfect"
  - You can always refine later
  - Focus on functionality first
- **Skipping Testing**
  - Test each component individually
  - Don't wait until the end
  - Use print statements liberally
- **Poor Time Management**
  - Work a little each day
  - Don't leave features for last minute
  - Account for integration time

# Project Planning - Assessment Criteria Review

**Individual Component (70%):**

- Core Functionality: 25%
- 3 Chosen Features: 30%
- Enhancement: 5%
- Code Quality: 5%
- Documentation: 5%

**Team Component (30%):**

- Team Feature: 15%
- Collaboration: 10%
- Presentation: 5%

# Review

- The capstone has 4 components: core, 3 features, enhancement, and optional team
- Everyone builds the same foundation but creates unique applications
- Feature selection should balance interest with skill level
- Success comes from steady progress and good planning
- The project is both a learning experience and portfolio piece

# Questions?