APPENDIX 1.1:

Subject: Help ZYX Inc. Streamline Marketing Efforts with Automation

Hi John,

I came across ZYX Inc. and was impressed with the innovative approach you're taking in the [industry/field]. As a Marketing Manager, you understand the importance of maximizing your team's productivity and driving revenue through efficient marketing strategies.

That's where our marketing automation tools come in. Our SaaS solution empowers businesses like yours to automate and optimize marketing campaigns, leading to increased conversions and revenue growth.

I'd love to discuss how our platform can help ZYX Inc.:

- Save time and resources by automating repetitive tasks
- Enhance customer engagement with personalized campaigns
- Gain valuable insights into marketing performance

If you're interested in exploring ways to take your marketing efforts to the next level, let's schedule a quick call. I'd be happy to walk you through a personalized demo and answer any questions you may have.

Looking forward to hearing from you, John!

Best regards,

APPENDIX 1.2:

```
<!-- HTML -->
<html>
<head>
	<title>Calculator</title>
	<link rel="stylesheet" href="style.css">
</head>
<body>
	<h1>Calculator</h1>
	<form>
		<input type="text" id="display" readonly>
		<table>
			<tr>
				<td><button onclick="press('C')">C</button></td>
				<td><button onclick="press('/')">/</button></td>
				<td><button onclick="press('*')">*</button></td>
				<td><button onclick="press('-')">-</button></td>
			</tr>
			<tr>
				<td><button onclick="press('7')">7</button></td>
				<td><button onclick="press('8')">8</button></td>
				<td><button onclick="press('9')">9</button></td>
				<td><button onclick="press('+')">+</button></td>
			</tr>
			<tr>
				<td><button onclick="press('4')">4</button></td>
				<td><button onclick="press('5')">5</button></td>
				<td><button onclick="press('6')">6</button></td>
				<td><button onclick="press('=')">=</button></td>
			</tr>
			<tr>
				<td><button onclick="press('1')">1</button></td>
				<td><button onclick="press('2')">2</button></td>
				<td><button onclick="press('3')">3</button></td>
				<td><button onclick="press('0')">0</button></td>
			</tr>
			<tr>
				<td colspan="4"><button
onclick="press('.')">.</button></td>
			</tr>
		</table>
	</form>
	<script src="script.js"></script>
</body>
</html>
```

```
<!-- CSS (in style.css file) -->
body {
      font-family: Arial, sans-serif;
}

table {
      border-collapse: collapse;
}

button {
      width: 50px;
      height: 50px;
      font-size: 24px;
      margin: 5px;
}

#display {
      width: 250px;
      height: 50px;
      font-size: 36px;
      padding: 10px;
      border: 1px solid #ccc;
}

<!-- JavaScript (in script.js file) -->
let display = document.getElementById('display');
let currentNumber = '';
let previousNumber = '';
let operation = '';

function press(value) {
      if (value === 'C') {
            currentNumber = '';
            previousNumber = '';
            operation = '';
            display.value = '0';
      } else if (value === '=') {
            calculate();
      } else if (value === '.') {
            if (!currentNumber.includes('.')) {
                  currentNumber += '.';
            }
      } else if (value === '/' || value === '*' || value === '-' ||
value === '+') {
            operation = value;
```

```
            previousNumber = currentNumber;
            currentNumber = '';
      } else {
            currentNumber += value;
      }
      display.value = currentNumber;
}

function calculate() {
      let result = 0;
      switch (operation) {
            case '/':
                  result = parseFloat(previousNumber) /
parseFloat(currentNumber);
                  break;
            case '*':
                  result = parseFloat(previousNumber) *
parseFloat(currentNumber);
                  break;
            case '-':
                  result = parseFloat(previousNumber) -
parseFloat(currentNumber);
                  break;
            case '+':
                  result = parseFloat(previousNumber) +
parseFloat(currentNumber);
                  break;
      }
      currentNumber = result.toString();
      previousNumber = '';
      operation = '';
      display.value = currentNumber;
}
```

APPENDIX 1.3:

```
class ChessGame:
    def __init__(self):
        self.board = self.initialize_board()
        self.current_player = "white"

    def initialize_board(self):
        # Initialize an 8x8 board with pieces in starting positions
        board = [["" for _ in range(8)] for _ in range(8)]
        # Add pawns
        for i in range(8):
            board[1][i] = "white_pawn"
            board[6][i] = "black_pawn"
        # Add other pieces
        for color in ["white", "black"]:
            for piece, position in [
                ("rook", (0, 0)),
                ("knight", (0, 1)),
                ("bishop", (0, 2)),
                ("queen", (0, 3)),
                ("king", (0, 4)),
                ("bishop", (0, 5)),
                ("knight", (0, 6)),
                ("rook", (0, 7)),
            ]:
                board[position[0]][position[1]] = f"{color}_{piece}"
        return board

    def print_board(self):
        print("  a b c d e f g h")
        for i, row in enumerate(self.board):
            print(f"{i+1} {' '.join(row)}")

    def make_move(self, start, end):
        # Check if move is valid (piece exists, move is allowed, etc.)
        # For simplicity, assume moves are always valid
        piece = self.board[start[0]][start[1]]
        self.board[start[0]][start[1]] = ""
        self.board[end[0]][end[1]] = piece
        self.current_player = "black" if self.current_player == "white"
else "white"

    def play_game(self):
        while True:
            self.print_board()
```

```
            start = input(f"Player {self.current_player}, enter start
position (e.g., 'e2'): ")
            end = input(f"Enter end position (e.g., 'e4'): ")
            start = (int(start[1]) - 1, ord(start[0]) - 97)
            end = (int(end[1]) - 1, ord(end[0]) - 97)
            self.make_move(start, end)

game = ChessGame()
game.play_game()
```

APPENDIX 2.1:

Hi John,

I hope this email finds you well.

My name is [Your Name] and I work at [Your Company], a marketing automation platform that helps businesses like ZYX Inc. optimize their marketing efforts and drive better results.

I came across ZYX Inc.'s recent campaign on [mention specific platform, e.g., Facebook Ads] promoting [mention specific product/service]. It looked really engaging!

At [Your Company], we understand that Marketing Managers like yourself wear many hats. That's why we offer tools that can automate repetitive tasks, personalize email campaigns, and track campaign performance in one central location.

For example, with our software, you could easily:

- **Free up valuable time** by automating email sequences and social media scheduling.
- **Boost lead generation** with personalized landing pages and targeted nurture campaigns.
- **Gain valuable insights** into your marketing performance with detailed analytics and reporting.

As a quick win, I'd be happy to share a free downloadable template for creating high-converting email workflows. Would you be interested in receiving a copy?

Let me know if you have a few minutes to chat about how [Your Company] can help ZYX Inc. achieve its marketing goals.

Thanks,

APPENDIX 2.2:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Simple Calculator</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Simple Calculator</h1>
    <div class="calculator">
        <input type="text" id="display" readonly>
        <div class="buttons">
            <button onclick="appendNumber('7')">7</button>
            <button onclick="appendNumber('8')">8</button>
            <button onclick="appendNumber('9')">9</button>
            <button onclick="appendOperator('/')">/</button>
            <button onclick="appendNumber('4')">4</button>
            <button onclick="appendNumber('5')">5</button>
            <button onclick="appendNumber('6')">6</button>
            <button onclick="appendOperator('*')">*</button>
            <button onclick="appendNumber('1')">1</button>
            <button onclick="appendNumber('2')">2</button>
            <button onclick="appendNumber('3')">3</button>
            <button onclick="appendOperator('-')">-</button>
            <button onclick="appendNumber('0')">0</button>
            <button onclick="clearDisplay()">C</button>
            <button onclick="calculate()">=</button>
            <button onclick="appendOperator('+')">+</button>
        </div>
    </div>
    <script src="script.js"></script>
</body>
</html>



body {
    font-family: Arial, sans-serif;
}

.calculator {
    width: 300px;
```

```
    margin: 0 auto;
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 10px;
}

#display {
    width: 100%;
    height: 50px;
    border: none;
    font-size: 20px;
    text-align: right;
    padding: 5px;
}

.buttons {
    display: flex;
    flex-wrap: wrap;
    gap: 5px;
}

button {
    width: 47%;
    height: 50px;
    border: none;
    font-size: 18px;
    cursor: pointer;
}

button:hover {
    background-color: #eee;
}



let currentNumber = "";
let previousNumber = "";
let operator = "";

function appendNumber(number) {
  currentNumber += number;
  document.getElementById("display").value = currentNumber;
}

function appendOperator(op) {
  previousNumber = currentNumber;
  currentNumber = "";
```

```
  operator = op;
}

function clearDisplay() {
  currentNumber = "";
  previousNumber = "";
  operator = "";
  document.getElementById("display").value = "";
}

function calculate() {
  let result = 0;
  const prevNum = parseFloat(previousNumber);
  const currNum = parseFloat(currentNumber);

  if (operator === "+") {
    result = prevNum + currNum;
  } else if (operator === "-") {
    result = prevNum - currNum;
  } else if (operator === "*") {
    result = prevNum * currNum;
  } else if (operator === "/") {
    if (currNum === 0) {
      alert("Error: Division by zero");
      return;
    }
    result = prevNum / currNum;
  }

  currentNumber = result.toString();
  operator = "";
  previousNumber = "";
  document.getElementById("display").value = currentNumber;
}
```

APPENDIX 2.3:

```
class Piece:
  def __init__(self, color, position):
    self.color = color
    self.position = position

  def __str__(self):
    return self.color[0] + self.symbol

class Board:
  def __init__(self):
    self.board = [[None for _ in range(8)] for _ in range(8)]
    self.setup_board()

  def setup_board(self):
    # Place pieces in starting positions (modify as needed)
    for i in range(8):
      self.board[1][i] = Piece("white", (1, i))
      self.board[6][i] = Piece("black", (6, i))
    self.board[0][0], self.board[0][7] = Piece("white", (0, 0)),
Piece("white", (0, 7))
    self.board[0][1], self.board[0][6] = Piece("white", (0, 1)),
Piece("white", (0, 6))
    self.board[7][0], self.board[7][7] = Piece("black", (7, 0)),
Piece("black", (7, 7))
    self.board[7][1], self.board[7][6] = Piece("black", (7, 1)),
Piece("black", (7, 6))

  def get_piece(self, position):
    x, y = position
    return self.board[x][y]

  def move_piece(self, piece, new_position):
    x, y = piece.position
    self.board[x][y] = None
    self.board[new_position[0]][new_position[1]] = piece
    piece.position = new_position

  def __str__(self):
    # Display the board (modify for better formatting)
    output = ""
    for row in self.board:
      for piece in row:
        output += str(piece) + " "
      output += "\n"
```

```
    return output

def is_valid_move(board, piece, new_position):
  # Implement logic to check valid moves based on piece type
  # Consider basic move patterns, attacks, and edge cases
  # This is a simplified example, needs extension for all pieces
  x, y = piece.position
  dx, dy = new_position[0] - x, new_position[1] - y
  if piece.color == "white":
    return dy > 0 and abs(dx) <= abs(dy)  # White pawns can only move
forward
  else:
    return dy < 0 and abs(dx) <= abs(dy)  # Black pawns can only move
down

def get_user_input():
  while True:
    try:
      position = input("Enter piece position (e.g., e4): ").lower()
      x, y = ord(position[0]) - 97, int(position[1]) - 1
      if 0 <= x <= 7 and 0 <= y <= 7:
        return (x, y)
      else:
        print("Invalid position. Please try again.")
    except ValueError:
      print("Invalid input. Please enter a valid position.")

def main():
  board = Board()
  current_turn = "white"

  while True:
    print(board)
    print(f"{current_turn}'s turn")

    piece_position = get_user_input()
    piece = board.get_piece(piece_position)

    if not piece or piece.color != current_turn:
      print("Invalid selection. Please choose your piece.")
      continue

    new_position = get_user_input()

    if not is_valid_move(board, piece, new_position):
      print("Invalid move. Please try again.")
```

```
      continue

    board.move_piece(piece, new_position)

    # Check for check and checkmate (not implemented in this example)

    current_turn = "black" if current_turn == "white" else "white"

if __name__ == "__main__":
  main()
```

APPENDIX 3.1:

Subject: Elevate Your Marketing Strategy with Cutting-Edge Automation Tools

Hi John,

I hope this email finds you well! My name is [Your Name], and I'm reaching out as a Sales Development Representative at [Your Company], a leading provider of innovative marketing automation solutions.

I came across your profile while researching innovative marketing leaders in the industry, and I was particularly impressed by your role as Marketing Manager at ZYX Inc. Your dedication to driving impactful strategies undoubtedly aligns with our mission to empower marketing teams to achieve their goals efficiently and effectively.

At [Your Company], we specialize in providing cutting-edge marketing automation tools designed to streamline workflows, enhance customer engagement, and drive measurable results. Our platform offers a comprehensive suite of features, including personalized email campaigns, lead scoring, advanced analytics, and much more.

I believe that ZYX Inc. could greatly benefit from leveraging our automation solutions to optimize your marketing efforts, increase ROI, and ultimately drive growth for your organization. I'd love the opportunity to discuss how our platform can address your specific needs and help you achieve your objectives.

Are you available for a brief call sometime this week to explore further? Please let me know a convenient time for you, or feel free to schedule directly using the link below:

[Insert Meeting Scheduling Link]

I'm looking forward to the possibility of collaborating with you and ZYX Inc. If you have any questions or would like more information in the meantime, please don't hesitate to reach out.

Warm regards,

APPENDIX 3.2:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Simple Calculator</title>
    <style>
        input[type="button"] {
            width: 50px;
            height: 50px;
            font-size: 20px;
        }
        #display {
            width: 200px;
            height: 50px;
            font-size: 24px;
        }
    </style>
</head>
<body>
    <h2>Simple Calculator</h2>
    <input type="text" id="display" disabled><br>
    <input type="button" value="1" onclick="appendToDisplay('1')">
    <input type="button" value="2" onclick="appendToDisplay('2')">
    <input type="button" value="3" onclick="appendToDisplay('3')">
    <input type="button" value="+" onclick="appendToDisplay('+')"><br>
    <input type="button" value="4" onclick="appendToDisplay('4')">
    <input type="button" value="5" onclick="appendToDisplay('5')">
    <input type="button" value="6" onclick="appendToDisplay('6')">
    <input type="button" value="-" onclick="appendToDisplay('-')"><br>
    <input type="button" value="7" onclick="appendToDisplay('7')">
    <input type="button" value="8" onclick="appendToDisplay('8')">
    <input type="button" value="9" onclick="appendToDisplay('9')">
    <input type="button" value="*" onclick="appendToDisplay('*')"><br>
    <input type="button" value="C" onclick="clearDisplay()">
    <input type="button" value="0" onclick="appendToDisplay('0')">
    <input type="button" value="/" onclick="appendToDisplay('/')">
    <input type="button" value="=" onclick="calculate()">

    <script>
        function appendToDisplay(value) {
            document.getElementById('display').value += value;
        }
```

```
        function clearDisplay() {
            document.getElementById('display').value = '';
        }

        function calculate() {
            try {
                let result =
eval(document.getElementById('display').value);
                document.getElementById('display').value = result;
            } catch (error) {
                document.getElementById('display').value = 'Error';
            }
        }
    </script>
</body>
</html>
```

APPENDIX 3.3:

```
import chess

def print_board(board):
    print(board)

def get_move(board):
    while True:
        try:
            move = input("Enter your move (e.g., e2e4): ")
            move = chess.Move.from_uci(move)
            if move in board.legal_moves:
                return move
            else:
                print("Invalid move! Try again.")
        except ValueError:
            print("Invalid move format! Try again.")

def main():
    board = chess.Board()

    print("Welcome to Python Chess!")
    print_board(board)

    while not board.is_game_over():
        if board.turn == chess.WHITE:
            print("White's turn:")
        else:
            print("Black's turn:")

        move = get_move(board)
        board.push(move)

        print_board(board)

    result = board.result()
    if result == "1-0":
        print("White wins!")
    elif result == "0-1":
        print("Black wins!")
    else:
        print("It's a draw!")

if __name__ == "__main__":
    main()
```