

# Drawing Constellations:

## HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exosky Drawing App</title>
  <link rel="stylesheet" href="styles.css">
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&dis
play=swap" rel="stylesheet">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/js/all.min.js"
></script>
</head>
<body>
  <div class="container">
    <div class="drawing-section">
      <canvas id="drawingCanvas" width="500" height="400"></canvas>
      <div class="controls">
        <input type="color" id="colorPicker" value="#ffffff">
        <input type="range" id="brushSize" min="1" max="20" value="2">
        <span id="brushSizeValue">2</span>
        <button id="undoBtn">Undo</button>
        <button id="redoBtn">Redo</button>
        <button id="clearBtn">Clear</button>
        <button id="finishBtn">Finish drawing Constellation</button>
      </div>
      <div id="descriptionBox">
        <p>Describe your Constellation:</p>
```

```

        <input type="text" id="exoskyDescription">
        <button id="submitDescription">Submit</button>
    </div>
</div>
<div id="messageBox" class="hidden">
    <h2>My Constellation</h2>
    <p id="message"></p>
    <div id="shareOptions">
        <h3>Share your Constellation:</h3>
        <button id="shareTwitter" class="share-btn"><i class="fab
fa-twitter"></i> Twitter</button>
        <button id="shareFacebook" class="share-btn"><i class="fab
fa-facebook-f"></i> Facebook</button>
        <button id="shareLinkedIn" class="share-btn"><i class="fab
fa-linkedin-in"></i> LinkedIn</button>
        <button id="copyLink" class="share-btn"><i class="fas
fa-link"></i> Copy Link</button>
    </div>
</div>
</div>
<script src="script.js"></script>
</body>
</html>

```

## CSS Code:

```

body {
    font-family: 'Roboto', sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}

```

```
    background-color: #1a1a2e;
    color: #ffffff;
}
.container {
    display: flex;
    gap: 20px;
    align-items: flex-start;
}
.drawing-section {
    text-align: center;
}
canvas {
    border: 2px solid #4a4e69;
    cursor: crosshair;
    background-color: #0f0f1b;
}
.controls {
    margin-top: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
}
button {
    margin: 5px;
    padding: 8px 12px;
    background-color: #4361ee;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
}
button:hover {
    background-color: #3a0ca3;
```

```
}
#colorPicker {
    margin-right: 10px;
}
#brushSize {
    width: 100px;
    margin: 0 10px;
}
#brushSizeValue {
    min-width: 20px;
    display: inline-block;
}
#descriptionBox {
    display: none;
    margin-top: 20px;
}
#exoskyDescription {
    padding: 5px;
    margin-right: 5px;
    border-radius: 3px;
    border: 1px solid #4a4e69;
    background-color: #0f0f1b;
    color: #ffffff;
}
#messageBox {
    width: 250px;
    padding: 20px;
    background-color: #16213e;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.3);
}
#messageBox.hidden {
    display: none;
}
#messageBox h2 {
```

```
    color: #e2e8f0;
    margin-top: 0;
    font-style: italic;
}
#message {
    font-size: 18px;
    color: #4cc9f0;
    font-weight: bold;
    line-height: 1.4;
}
#shareOptions {
    margin-top: 20px;
}
#shareOptions h3 {
    margin-bottom: 10px;
}
.share-btn {
    display: inline-flex;
    align-items: center;
    justify-content: center;
    width: 100%;
    margin-bottom: 5px;
    padding: 8px;
    font-size: 14px;
}
.share-btn i {
    margin-right: 5px;
}
#shareTwitter { background-color: #1DA1F2; }
#shareFacebook { background-color: #4267B2; }
#shareLinkedIn { background-color: #0077B5; }
#copyLink { background-color: #6c757d; }
```

**JavaScript Code:**

```
const canvas = document.getElementById('drawingCanvas');
const ctx = canvas.getContext('2d');
const colorPicker = document.getElementById('colorPicker');
const brushSize = document.getElementById('brushSize');
const brushSizeValue = document.getElementById('brushSizeValue');
const undoBtn = document.getElementById('undoBtn');
const redoBtn = document.getElementById('redoBtn');
const clearBtn = document.getElementById('clearBtn');
const finishBtn = document.getElementById('finishBtn');
const descriptionBox = document.getElementById('descriptionBox');
const exoskyDescription = document.getElementById('exoskyDescription');
const submitDescription = document.getElementById('submitDescription');
const messageBox = document.getElementById('messageBox');
const message = document.getElementById('message');
const shareTwitter = document.getElementById('shareTwitter');
const shareFacebook = document.getElementById('shareFacebook');
const shareLinkedIn = document.getElementById('shareLinkedIn');
const copyLink = document.getElementById('copyLink');
```

```
let isDrawing = false;
let lastX = 0;
let lastY = 0;
let color = '#ffffff';
let lineWidth = 2;
let drawingStates = [];
let currentStateIndex = -1;
```

```
// Set initial canvas background to dark blue
ctx.fillStyle = '#0f0f1b';
ctx.fillRect(0, 0, canvas.width, canvas.height);
saveState();
```

```
function startDrawing(e) {
  isDrawing = true;
  [lastX, lastY] = [e.offsetX, e.offsetY];
```

```
}
```

```
function draw(e) {  
    if (!isDrawing) return;  
    ctx.strokeStyle = color;  
    ctx.lineWidth = lineWidth;  
    ctx.lineCap = 'round';  
    ctx.beginPath();  
    ctx.moveTo(lastX, lastY);  
    ctx.lineTo(e.offsetX, e.offsetY);  
    ctx.stroke();  
    [lastX, lastY] = [e.offsetX, e.offsetY];  
}
```

```
function stopDrawing() {  
    if (isDrawing) {  
        isDrawing = false;  
        saveState();  
    }  
}
```

```
function saveState() {  
    currentStateIndex++;  
    drawingStates = drawingStates.slice(0, currentStateIndex);  
    drawingStates.push(canvas.toDataURL());  
}
```

```
function undo() {  
    if (currentStateIndex > 0) {  
        currentStateIndex--;  
        restoreState();  
    }  
}
```

```
function redo() {
```

```

    if (currentStateIndex < drawingStates.length - 1) {
        currentStateIndex++;
        restoreState();
    }
}

```

```

function restoreState() {
    const img = new Image();
    img.src = drawingStates[currentStateIndex];
    img.onload = () => {
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        ctx.drawImage(img, 0, 0);
    };
}

```

```

function clearCanvas() {
    ctx.fillStyle = '#0f0f1b';
    ctx.fillRect(0, 0, canvas.width, canvas.height);
    saveState();
}

```

```

function shareOnSocialMedia(platform) {
    const description = encodeURIComponent(message.textContent);
    const imageData = canvas.toDataURL('image/png');
    let url;

    switch (platform) {
        case 'twitter':
            url =
`https://twitter.com/intent/tweet?text=${description}&url=${encodeURIComponent(window.location.href)}`;
            break;
        case 'facebook':

```



```

        url =
`https://www.facebook.com/sharer/sharer.php?u=${encodeURIComponent(
window.location.href)}`;
        break;
        case 'linkedin':
            url =
`https://www.linkedin.com/shareArticle?mini=true&url=${encodeURIComponent(
window.location.href)}&title=${encodeURIComponent('My Exosky
Drawing')}&summary=${description}`;
            break;
    }

    window.open(url, '_blank');
}

```

```

function copyShareableLink() {
    const imageData = canvas.toDataURL('image/png');
    const dummyLink = `${window.location.origin}/share?id=${Date.now()}`;
    navigator.clipboard.writeText(dummyLink).then(() => {
        alert('Shareable link copied to clipboard!');
    }).catch(err => {
        console.error('Failed to copy link: ', err);
    });
}

```

```

canvas.addEventListener('mousedown', startDrawing);
canvas.addEventListener('mousemove', draw);
canvas.addEventListener('mouseup', stopDrawing);
canvas.addEventListener('mouseout', stopDrawing);

```

```

colorPicker.addEventListener('input', (e) => {
    color = e.target.value;
});

```

```

brushSize.addEventListener('input', (e) => {

```

```

    lineWidth = e.target.value;
    brushSizeValue.textContent = lineWidth;
  });

  undoBtn.addEventListener('click', undo);
  redoBtn.addEventListener('click', redo);
  clearBtn.addEventListener('click', clearCanvas);

  finishBtn.addEventListener('click', () => {
    descriptionBox.style.display = 'block';
    messageBox.classList.add('hidden');
  });

  submitDescription.addEventListener('click', () => {
    const description = exoskyDescription.value;
    message.textContent = `What my exosky actually depicts:
    ${description}`;
    descriptionBox.style.display = 'none';
    messageBox.classList.remove('hidden');
    exoskyDescription.value = "";
  });

  shareTwitter.addEventListener('click', () => shareOnSocialMedia('twitter'));
  shareFacebook.addEventListener('click', () =>
  shareOnSocialMedia('facebook'));
  shareLinkedIn.addEventListener('click', () =>
  shareOnSocialMedia('linkedin'));
  copyLink.addEventListener('click', copyShareableLink);

```

## Drawing Exosky:

### HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exosky Drawing App</title>
  <link rel="stylesheet" href="styles.css">
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&dis
play=swap" rel="stylesheet">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/js/all.min.js"
></script>
</head>
<body>
  <div class="container">
    <div class="drawing-section">
      <canvas id="drawingCanvas" width="500" height="400"></canvas>
      <div class="controls">
        <input type="color" id="colorPicker" value="#ffffff">
        <input type="range" id="brushSize" min="1" max="20" value="2">
        <span id="brushSizeValue">2</span>
        <button id="undoBtn">Undo</button>
        <button id="redoBtn">Redo</button>
        <button id="clearBtn">Clear</button>
        <button id="finishBtn">Finish drawing Exosky</button>
      </div>
      <div id="descriptionBox">
        <p>Describe your exosky:</p>
        <input type="text" id="exoskyDescription">
        <button id="submitDescription">Submit</button>
      </div>
    </div>
    <div id="messageBox" class="hidden">
      <h2>My Exosky</h2>
```

```

    <p id="message"></p>
    <div id="shareOptions">
      <h3>Share your Exosky:</h3>
      <button id="shareTwitter" class="share-btn"><i class="fab
fa-twitter"></i> Twitter</button>
      <button id="shareFacebook" class="share-btn"><i class="fab
fa-facebook-f"></i> Facebook</button>
      <button id="shareLinkedIn" class="share-btn"><i class="fab
fa-linkedin-in"></i> LinkedIn</button>
      <button id="copyLink" class="share-btn"><i class="fas
fa-link"></i> Copy Link</button>
    </div>
  </div>
</div>
<script src="script.js"></script>
</body>
</html>

```

## CSS Code:

```

body {
  font-family: 'Roboto', sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  background-color: #1a1a2e;
  color: #ffffff;
}
.container {
  display: flex;
  gap: 20px;

```

```
    align-items: flex-start;
}
.drawing-section {
    text-align: center;
}
canvas {
    border: 2px solid #4a4e69;
    cursor: crosshair;
    background-color: #0f0f1b;
}
.controls {
    margin-top: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
}
button {
    margin: 5px;
    padding: 8px 12px;
    background-color: #4361ee;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
}
button:hover {
    background-color: #3a0ca3;
}
#colorPicker {
    margin-right: 10px;
}
#brushSize {
    width: 100px;
```

```
    margin: 0 10px;
}
#brushSizeValue {
    min-width: 20px;
    display: inline-block;
}
#descriptionBox {
    display: none;
    margin-top: 20px;
}
#exoskyDescription {
    padding: 5px;
    margin-right: 5px;
    border-radius: 3px;
    border: 1px solid #4a4e69;
    background-color: #0f0f1b;
    color: #ffffff;
}
#messageBox {
    width: 250px;
    padding: 20px;
    background-color: #16213e;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.3);
}
#messageBox.hidden {
    display: none;
}
#messageBox h2 {
    color: #e2e8f0;
    margin-top: 0;
    font-style: italic;
}
#message {
    font-size: 18px;
```

```

    color: #4cc9f0;
    font-weight: bold;
    line-height: 1.4;
}
#shareOptions {
    margin-top: 20px;
}
#shareOptions h3 {
    margin-bottom: 10px;
}
.share-btn {
    display: inline-flex;
    align-items: center;
    justify-content: center;
    width: 100%;
    margin-bottom: 5px;
    padding: 8px;
    font-size: 14px;
}
.share-btn i {
    margin-right: 5px;
}
#shareTwitter { background-color: #1DA1F2; }
#shareFacebook { background-color: #4267B2; }
#shareLinkedIn { background-color: #0077B5; }
#copyLink { background-color: #6c757d; }

```

### JavaScript Code:

```

const canvas = document.getElementById('drawingCanvas');
const ctx = canvas.getContext('2d');
const colorPicker = document.getElementById('colorPicker');
const brushSize = document.getElementById('brushSize');
const brushSizeValue = document.getElementById('brushSizeValue');

```

```
const undoBtn = document.getElementById('undoBtn');
const redoBtn = document.getElementById('redoBtn');
const clearBtn = document.getElementById('clearBtn');
const finishBtn = document.getElementById('finishBtn');
const descriptionBox = document.getElementById('descriptionBox');
const exoskyDescription = document.getElementById('exoskyDescription');
const submitDescription = document.getElementById('submitDescription');
const messageBox = document.getElementById('messageBox');
const message = document.getElementById('message');
const shareTwitter = document.getElementById('shareTwitter');
const shareFacebook = document.getElementById('shareFacebook');
const shareLinkedIn = document.getElementById('shareLinkedIn');
const copyLink = document.getElementById('copyLink');
```

```
let isDrawing = false;
let lastX = 0;
let lastY = 0;
let color = '#ffffff';
let lineWidth = 2;
let drawingStates = [];
let currentStateIndex = -1;
```

```
// Set initial canvas background to dark blue
ctx.fillStyle = '#0f0f1b';
ctx.fillRect(0, 0, canvas.width, canvas.height);
saveState();
```

```
function startDrawing(e) {
  isDrawing = true;
  [lastX, lastY] = [e.offsetX, e.offsetY];
}
```

```
function draw(e) {
  if (!isDrawing) return;
  ctx.strokeStyle = color;
```



```

    ctx.lineWidth = lineWidth;
    ctx.lineCap = 'round';
    ctx.beginPath();
    ctx.moveTo(lastX, lastY);
    ctx.lineTo(e.offsetX, e.offsetY);
    ctx.stroke();
    [lastX, lastY] = [e.offsetX, e.offsetY];
}

function stopDrawing() {
    if (isDrawing) {
        isDrawing = false;
        saveState();
    }
}

function saveState() {
    currentStateIndex++;
    drawingStates = drawingStates.slice(0, currentStateIndex);
    drawingStates.push(canvas.toDataURL());
}

function undo() {
    if (currentStateIndex > 0) {
        currentStateIndex--;
        restoreState();
    }
}

function redo() {
    if (currentStateIndex < drawingStates.length - 1) {
        currentStateIndex++;
        restoreState();
    }
}

```

```
function restoreState() {
  const img = new Image();
  img.src = drawingStates[currentStateIndex];
  img.onload = () => {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    ctx.drawImage(img, 0, 0);
  };
}
```

```
function clearCanvas() {
  ctx.fillStyle = '#0f0f1b';
  ctx.fillRect(0, 0, canvas.width, canvas.height);
  saveState();
}
```

```
function shareOnSocialMedia(platform) {
  const description = encodeURIComponent(message.textContent);
  const imageData = canvas.toDataURL('image/png');
  let url;

  switch (platform) {
    case 'twitter':
      url =
`https://twitter.com/intent/tweet?text=${description}&url=${encodeURIComponent(window.location.href)}`;
      break;
    case 'facebook':
      url =
`https://www.facebook.com/sharer/sharer.php?u=${encodeURIComponent(window.location.href)}`;
      break;
    case 'linkedin':
      url =
`https://www.linkedin.com/shareArticle?mini=true&url=${encodeURIComponent(window.location.href)}`;
      break;
  }
}
```

```
nent(window.location.href)}&title=${encodeURIComponent('My Exosky  
Drawing')}&summary=${description}`;
```

```
    break;
```

```
  }
```

```
    window.open(url, '_blank');
```

```
  }
```

```
function copyShareableLink() {
```

```
    const imageData = canvas.toDataURL('image/png');
```

```
    const dummyLink = `${window.location.origin}/share?id=${Date.now()}`;
```

```
    navigator.clipboard.writeText(dummyLink).then(() => {
```

```
        alert('Shareable link copied to clipboard!');
```

```
    }).catch(err => {
```

```
        console.error('Failed to copy link: ', err);
```

```
    });
```

```
}
```

```
canvas.addEventListener('mousedown', startDrawing);
```

```
canvas.addEventListener('mousemove', draw);
```

```
canvas.addEventListener('mouseup', stopDrawing);
```

```
canvas.addEventListener('mouseout', stopDrawing);
```

```
colorPicker.addEventListener('input', (e) => {
```

```
    color = e.target.value;
```

```
});
```

```
brushSize.addEventListener('input', (e) => {
```

```
    lineWidth = e.target.value;
```

```
    brushSizeValue.textContent = lineWidth;
```

```
});
```

```
undoBtn.addEventListener('click', undo);
```

```
redoBtn.addEventListener('click', redo);
```

```
clearBtn.addEventListener('click', clearCanvas);
```

```
finishBtn.addEventListener('click', () => {  
  descriptionBox.style.display = 'block';  
  messageBox.classList.add('hidden');  
});
```

```
submitDescription.addEventListener('click', () => {  
  const description = exoskyDescription.value;  
  message.textContent = `What my constellation actually depicts:  
  ${description}`;  
  descriptionBox.style.display = 'none';  
  messageBox.classList.remove('hidden');  
  exoskyDescription.value = "";  
});
```

```
shareTwitter.addEventListener('click', () => shareOnSocialMedia('twitter'));  
shareFacebook.addEventListener('click', () =>  
  shareOnSocialMedia('facebook'));  
shareLinkedIn.addEventListener('click', () =>  
  shareOnSocialMedia('linkedin'));  
copyLink.addEventListener('click', copyShareableLink);
```