**AIM: Build Factory Design Pattern on the Model.**

Model chosen: Phishing Detection

// Used Factory design to build Builder Design

interface PhishingDetector {

    void detect();

}

class EmailPhishingDetector implements PhishingDetector {

    private String source;

    public EmailPhishingDetector(String source) {

        this.source = source;

    }

    public void detect() {

        // code for email phishing detection

        System.out.println("Email Phishing Attack Detector");

    }

}

class WebsitePhishingDetector implements PhishingDetector {

    private String source;

    public WebsitePhishingDetector(String source) {

        this.source = source;

    }

    public void detect() {

        // code for website phishing detection

        System.out.println("Website Phishing Attack Detector");

    }

}

class PhishingDetectorBuilder {

    private String type;

    private String source;

    public PhishingDetectorBuilder setType(String type) {

        this.type = type;

        return this;

    }

    public PhishingDetectorBuilder setSource(String source) {

        this.source = source;

        return this;

    }

    public PhishingDetector build() {

        if (type.equalsIgnoreCase("EMAIL")) {

            return new EmailPhishingDetector(source);

        } else if (type.equalsIgnoreCase("WEBSITE")) {

            return new WebsitePhishingDetector(source);

        }

        return null;

    }

}

// Test Class

public class PhishingAttackBuilder {

    public static void main(String[] args) {

        PhishingDetectorBuilder builder = new PhishingDetectorBuilder();

        PhishingDetector detector = builder

                            .setType("EMAIL")

                            .setSource("example@example.com")

                            .build();

        detector.detect();

    }

}

**Output:**

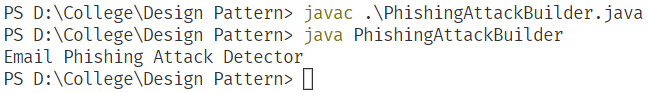
****

Figure 1: Output of the above program