

Yashotkarsh Work Report

16/02/2026 Monday

Competitor Research

Creating Report

Bynix Report

competitor : <https://www.webclickindia.com/>

Header : add portfolio and results

Footer : office address

Naybar : Graphical



Home Page : add animated elements right side instead of contact form and form toggle in a enquiry pop up button

WhatsApp sticky icon

About us : graphic animated

competitor : <https://www.konstantinfo.com/>

Header : contact drop down

Get a Quote-

- heading
- subheading
- tell about you :

Name



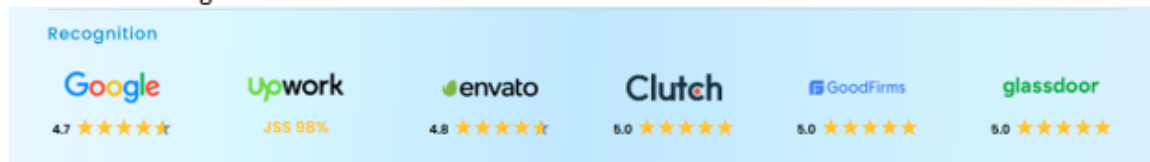
Blog : In blog write about the work that company do not in a different way like

- 20 Top Web Development Companies in India (2026)
- 10+ Apps like Airbnb 2026: Top Airbnb Alternatives
- Tinder Alternatives: Top Dating Apps like Tinder in 2026

that most probable easy to appear in search engines means topics that most common and related to company that solve business problems

competitor : <https://wpwebinfotech.com/>

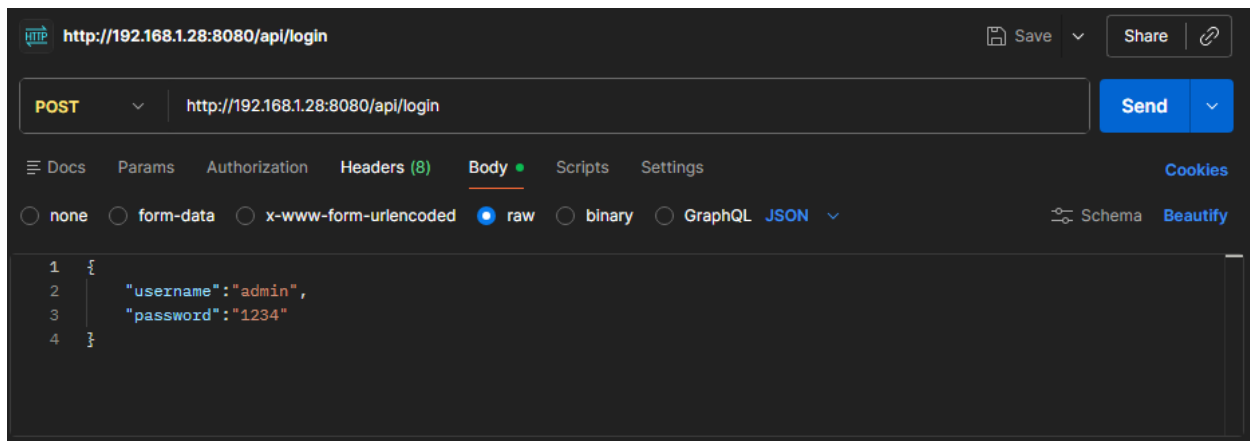
Footer : Recognition



FAQ :

- How many years of experience does the team have?
- Can you ensure quality in outsourced web development projects?
- As a leading web development outsourcing company, what are common engagement models you offer?
- Do you sign a Non-Disclosure Agreement?
- What method of communication should we use to communicate with your project development team?
- What web development standards do you follow?
- What is your process for Website Development?

Api Creation



Login Controller:

```
package com.example.demo.controller;

import com.example.demo.security.JwtUtil;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
```

```

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController
public class LoginController {
    private final JwtUtil jwtUtil;

    public LoginController(JwtUtil jwtUtil) {
        this.jwtUtil = jwtUtil;
    }

    @PostMapping("/api/login")
    public String login(@RequestBody Map<String, String> body) {

        String username = body.get("username");
        String password = body.get("password");

        if ("admin".equals(username) && "1234".equals(password)) {
            return jwtUtil.generateToken(username);
        }

        return "Invalid credentials";
    }
}

```

Trying to implement JWT Authentication

JwtUtil.java

```

package com.example.demo.security;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.security.Keys;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

import java.util.Date;

@Component
public class JwtUtil {

    @Value("${jwt.secret}")
    private String secret;

    @Value("${jwt.expiration}")
    private String expiration;

    public String generateToken(String username){
        return Jwts.builder()
            .setSubject(username)
            .setIssuedAt(new Date())
            .setExpiration(new Date(System.currentTimeMillis() +
expiration))
            .signWith(Keys.hmacShaKeyFor(secret.getBytes()),
SignatureAlgorithm.HS256)
            .compact();
    }
}

```

```

    }

    public String extractUsername(String token) {
        return Jwts.parserBuilder()
            .setSigningKey(secret.getBytes())
            .build()
            .parseClaimsJws(token)
            .getBody()
            .getSubject();
    }
}

```

JwtFilter.java

```

package com.example.demo.security;

import jakarta.servlet.FilterChain;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Component;
import org.springframework.web.filter.OncePerRequestFilter;

import java.io.IOException;
import java.util.ArrayList;

@Component
public class JwtFilter extends OncePerRequestFilter {

    private final JwtUtil jwtUtil;

    public JwtFilter(JwtUtil jwtUtil) {
        this.jwtUtil = jwtUtil;
    }

    @Override
    protected void doFilterInternal(HttpServletRequest request,
                                    HttpServletResponse response,
                                    FilterChain filterChain)
        throws ServletException, IOException {
        String header = request.getHeader("Authorization");
        if(header !=null && header.startsWith("Bearer ")){
            String token = header.substring(7);
            String username = null;
            try {
                username = jwtUtil.extractUsername(token);
            } catch (Exception e) {
                response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);
                return;
            }
            UsernamePasswordAuthenticationToken authToken =
                new UsernamePasswordAuthenticationToken(username, null,

```

```
new ArrayList<>());  
    SecurityContextHolder.getContext().setAuthentication(authToken);  
    }  
    filterChain.doFilter(request, response);  
    }  
}
```