Progress Report
Product-based Capstone Project

Project Name : **BodyBuddy**
Team ID : **CH2-PR633**
Team Member :

1. **(ML) M010BSY0667 - Luthfan Alifi Anwar (Active)**
2. **(ML) M193BSY1853 - Fetrianko Satriohaji (Active)**
3. **(MD) A355BSY2055 - Achmad Choiri Imroni (Active)**
4. **(CC) C102BSY3385 - Sergio Christian Winata (Active)**
5. **(CC) C002BSY3598 - Jordan Theo Immanuel Sihombing (Active)**
6. **(CC) C647BSY3894 - Yovinus Maehendra (Active)**
7. **(CC) C116BSY4366 - Ismail Saputra Sapareng (Active)**

Project Schedule (based on project plan):
Our Schedule from Project Plan, we made in Gantt chart format
Link :
https://docs.google.com/spreadsheets/d/1uVeM3Ob3sE8uAq6NvfY3qoKra5DncWS-/edit?usp=sharing&ouid=101324229037429284462&rtpof=true&sd=true

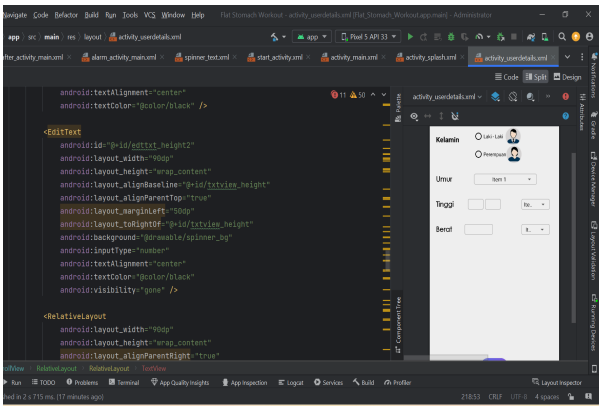
Project Progress Description:
Our team is already discussing how the application presented for the capstone will look like, including determining the features and functionalities needed. Our group's project focuses on creating an api for users to login and register, using Json Web Token(JWT). Creating the database using Cloud SQL. In addition to that overall goal, we are also completing our per-path progress such as designing the application in Figma (MD). We are currenting building the application itself, with per-path roles: Training a model from the datasets and converting into Android application-ready format in TensorFlow Lite (ML), coding application logics and interface of the Android application (MD), and building API endpoints for supporting the application (CC). In the future, we will pursue final stages of development which include testing activities, GitHub repositories building, presentation construction, and pitch deck video creation.

Progress Report
Product-based Capstone Project

If the complete capstone is 100%, tell us how many percent of your group's project has been completed. Please describe the reason for your percentage.
At the time of reporting, project progress should have reached at least 50% of the total project scope

The progress we have as described in the progress description according to the Gantt chart (link attached in this report) and the project plan, we are already done around that percentage.

Please attach your supporting evidence.

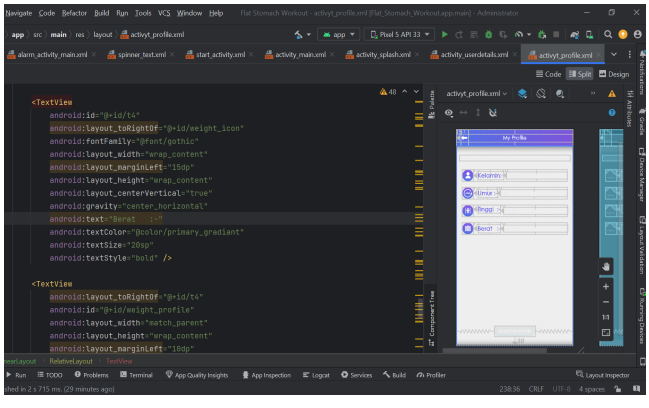


```python
# Initialize schemas
user_schema = UserSchema()
fitness_plan_schema = FitnessPlanSchema()

# Middleware for Token Authentication
def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.headers.get('Authorization')
        if not token:
            return jsonify({'message': 'Token is missing!'}), 401

        try:
            data = jwt.decode(token.split(" ")[1], app.config['SECRET_KEY'])
            current_user = User.query.filter_by(user_id=data['user_id']).first()
        except:
            return jsonify({'message': 'Token is invalid!'}), 401

        return f(current_user, *args, **kwargs)

    return decorated
```

```python
# Routes
@app.route('/api/users', methods=['POST'])
def register_user():
    data = request.json
    hashed_password = bcrypt.generate_password_hash(data['password']).decode('utf-8')
    new_user = User(username=data['username'], password_hash=hashed_password,
                    height=data['height'], weight=data['weight'], age=data['age'], gender=data['gender'])
    db.session.add(new_user)
    db.session.commit()
    return user_schema.jsonify(new_user)

@app.route('/api/login', methods=['POST'])
def login():
    auth = request.json
    user = User.query.filter_by(username=auth['username']).first()

    if user and bcrypt.check_password_hash(user.password_hash, auth['password']):
        token = jwt.encode({'user_id': user.user_id, 'exp': datetime.datetime.utcnow() + datetime.timedelta(days=1)},
                           app.config['SECRET_KEY'])
        return jsonify({'token': token.decode('UTF-8')})
    return jsonify({'message': 'Invalid username or password'}), 401

@app.route('/api/users/<int:user_id>', methods=['GET'])
@token_required
def get_user_profile(current_user, user_id):
    if current_user.user_id != user_id:
        return jsonify({'message': 'Unauthorized access'}), 403

    user = User.query.get(user_id)
    return user_schema.jsonify(user)
```